USE CASE

*red colour font means those are not implemented

Components required-

1. **Raspberry pi 4b**
2. **ESP8266 (NodeMCU)**
3. **Relay Switch**
4. **USB data cable**
5. **Jumper wires**

OS NEEDED

Raspberry Pi OS with desktop

Release date: March 4th 2021

Kernel version: 5.10

Size: 1,175MB

SD card

16GB/32GB

Preferable high speed sdcard

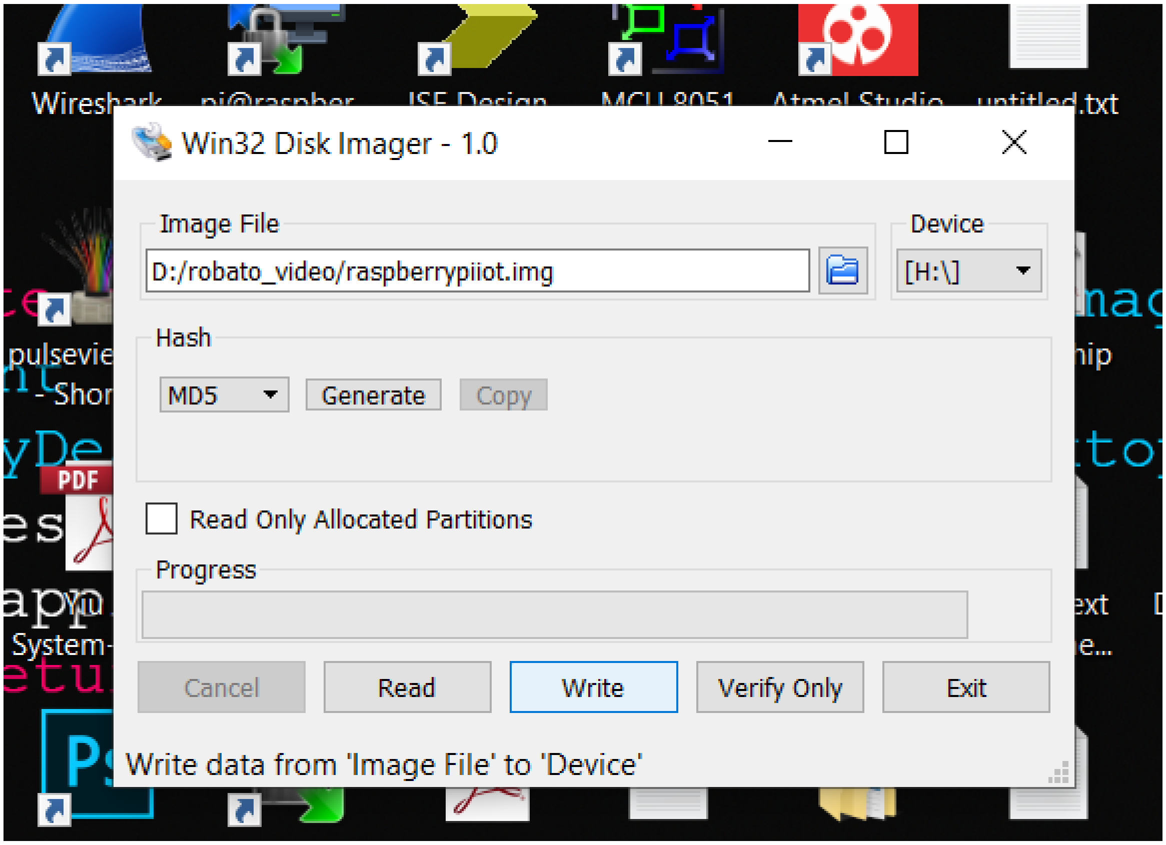
Download the my os image (6gb compressed file)

https://drive.google.com/file/d/1XpPOEP87ycJtb-3VWN1jG7g3xcPNyEvy/view?usp=sharing

- **After extracting the zipfile**
- **then download**
- **disk32 imager to install os to sdcard**

https://sourceforge.net/projects/win32diskimager/

- **insert sdcard on pc using sdcard reader,select the location of the extracted os image**
- **raspberrypiiot.img**
- **select MD5**
- **click on write**



Win32 Disk Imager - 1.0

Image File: D:/robato_video/raspberrypiiot.img

Device: [H:\]

Hash: MD5

□ Read Only Allocated Partitions

Progress

Write data from 'Image File' to 'Device'

1.DATABASE SETUP

- **Check the videolink and refer doc for reference**
- **Watch all the videos in 720p or 1080p only**

https://drive.google.com/file/d/1ioGUwRZF52lBJbPO1zuxdjH4e9j2qS4f/view?usp=sharing

- **Checking the SQL database**
- **Already installed with the os.img**
- **Type cmd**

```
sudo mysql -u root -p
```

- **Type password**

```
password
```

- **Database name**
- **SQL_GATEWAY_DATABASE**

- **4 Tables**

1.USERS_DATA –(USERNAME,PASSWORD,SECURITYKEY)

2. GATEWAY_CONFIG-(SSID,PASSWORD,GATEWAYIP)

3. ROOM_CONTROLBOX_APPLIANCE-

(DATE ,LOCATION ,CLIENT APPLIANCE , ON_TIME , OFF_TIME , TOTAL_ON_TIME)

4. CONTROLBOX_CONFIG

(ROOM ,DEVICE ,APPLIANCE1,APPLIANCE2, APPLIANCE3,APPLIANCE4)

- **TYPE CMD**

```
USE SQL_GATEWAY_DATABASE;
```

- **TYPE CMDS**

```
SELECT * FROM USERS_DATA;
SELECT * FROM GATEWAY_CONFIG;

SELECT * FROM ROOM_CONTROLBOX_APPLIANCE;

SELECT * FROM CONTROLBOX_CONFIG;
```

- **NEXT DELETE THE PREVIOUS DATAS FROM TABLE**

```
DELETE  FROM USERS_DATA;

DELETE  FROM GATEWAY_CONFIG;

DELETE  FROM ROOM_CONTROLBOX_APPLIANCE;

DELETE  FROM CONTROLBOX_CONFIG;
```

- **CHANGE THE ROOT PASSWORD AND SET YOUR OWN PASSWORD FOR SQL DATABASE**

```
SET PASSWORD FOR 'root'@'localhost' = PASSWORD('MY_NEW_PASSWORD');

FLUSH PRIVILEGES;
```

First customer will buy the product now below process is to be followed-

- **Mobile – Gateway**

  Set Wi-Fi credentials via Bluetooth (Secure Login) – First time default fixed password.

Step 1 - Customer will install app in his smartphone to configure and control devices.

Step 2- **Customer will plug and turn ON the gateway (raspberry pi). When gateway starts, it will open its Bluetooth, now customer will connect the Bluetooth of gateway using a default password known to customer, and go to the installed app and configure (Wi-Fi credential) the gateway.**

Step 3- Now gateway will connect with the wi-fi and obtain an ip address, and send the ip address to app using the Bluetooth.

Step 4- A python program will run a web server which will be accessed by the user app to configure the gateway in next step. Web page will be secured with a default id and password, which will be known to user.

- **BLUETOOTH CONFIGURATIONS TEST**

  VIDEO LINK-

  https://drive.google.com/file/d/1Web_4l0AD6IAzoAecvCfiAdqYdt_Qw1i/view?usp=sharing

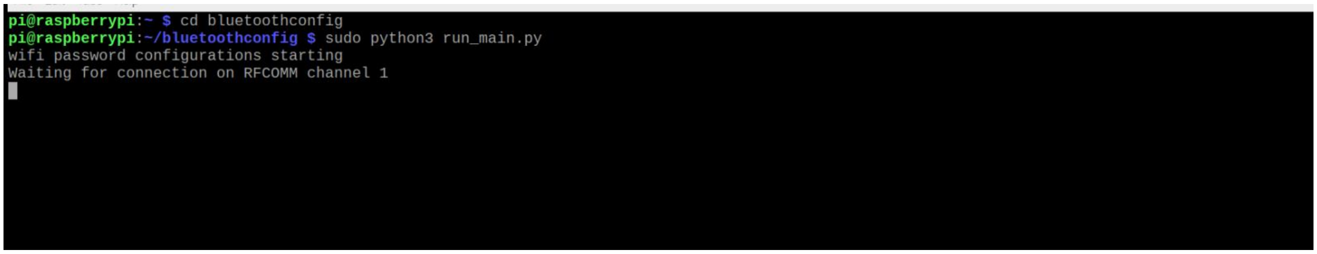
- **Bluetooth app1 link**

  https://res.cloudinary.com/di2vaxvhl/raw/upload/v1571090089/apk_Ubidots_source_code.zip

- **Procedure**
- **.Open the file containing run_main.py**
- **Type cmd**

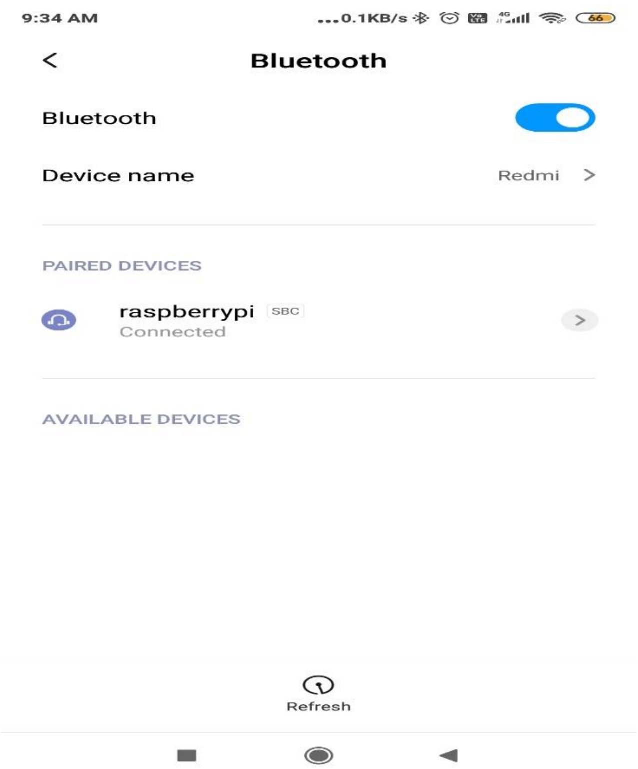
  cd bluetoothconfig

  sudo python3 run.py



```
pi@raspberrypi:~ $ cd bluetoothconfig
pi@raspberrypi:~/bluetoothconfig $ sudo python3 run_main.py
wifi password configurations starting
Waiting for connection on RFCOMM channel 1
█
```

- **Pairing of Bluetooth with raspberry pi must have completed early before ,now just connect mobile Bluetooth to raspi bluetooth**



Bluetooth ●

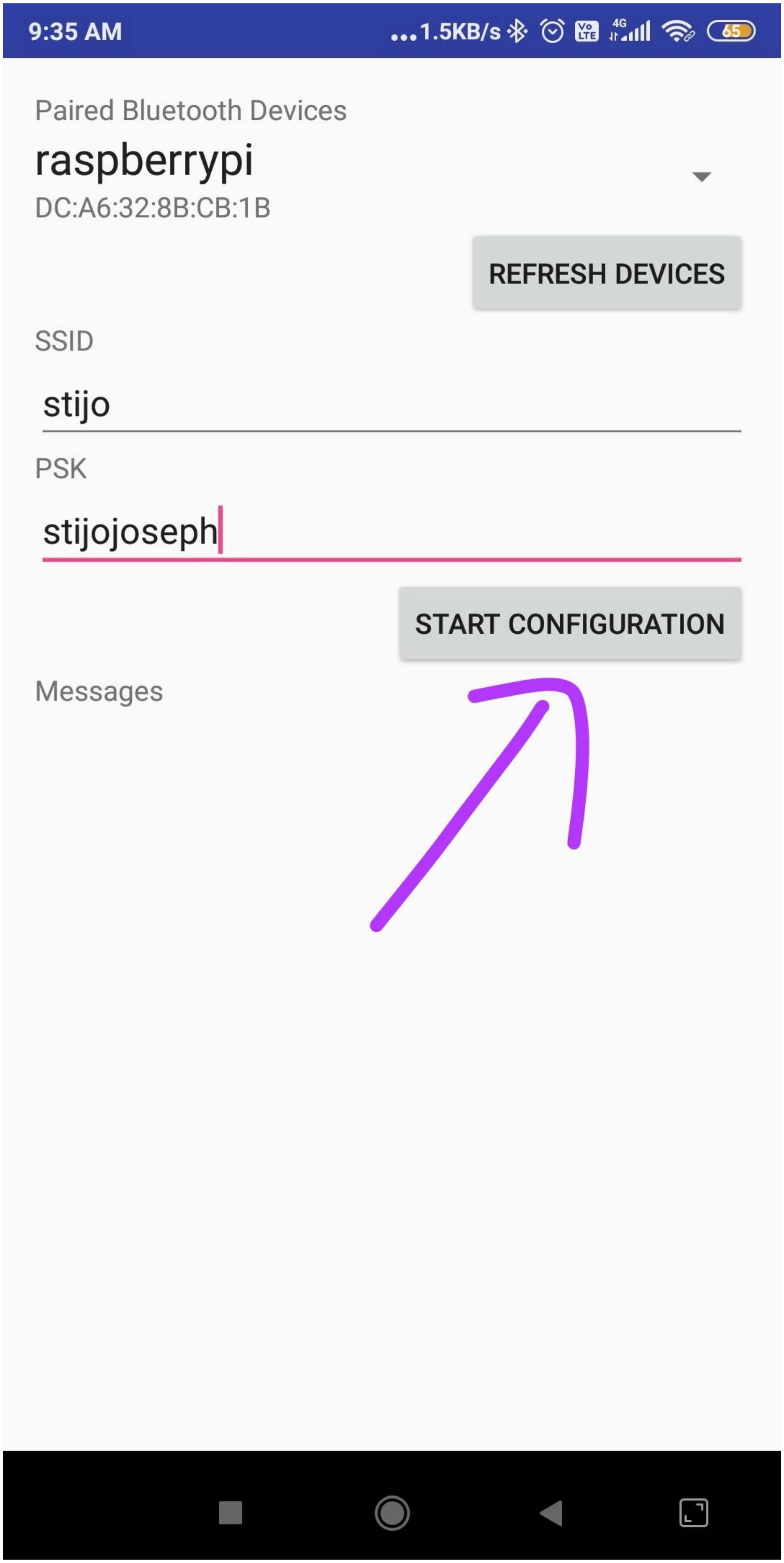
Device name Redmi >

PAIRED DEVICES

raspberrypi SBC
Connected >

AVAILABLE DEVICES

Open the app type your wifi ssid and password, from your phones portable hotspot settings or if you are using Router accordingly type it with out mistake



Paired Bluetooth Devices

raspberrypi

DC:A6:32:8B:CB:1B

SSID

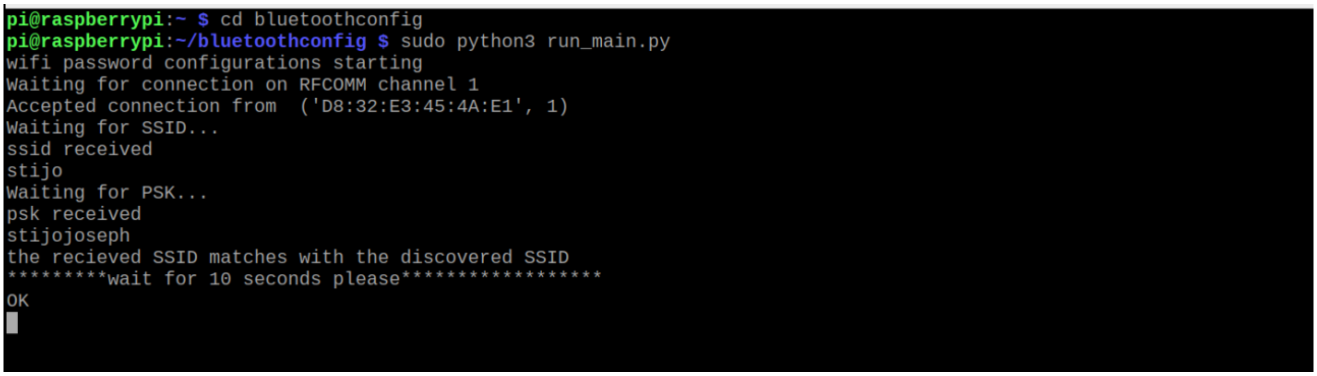
stijo

PSK

stijojoseph

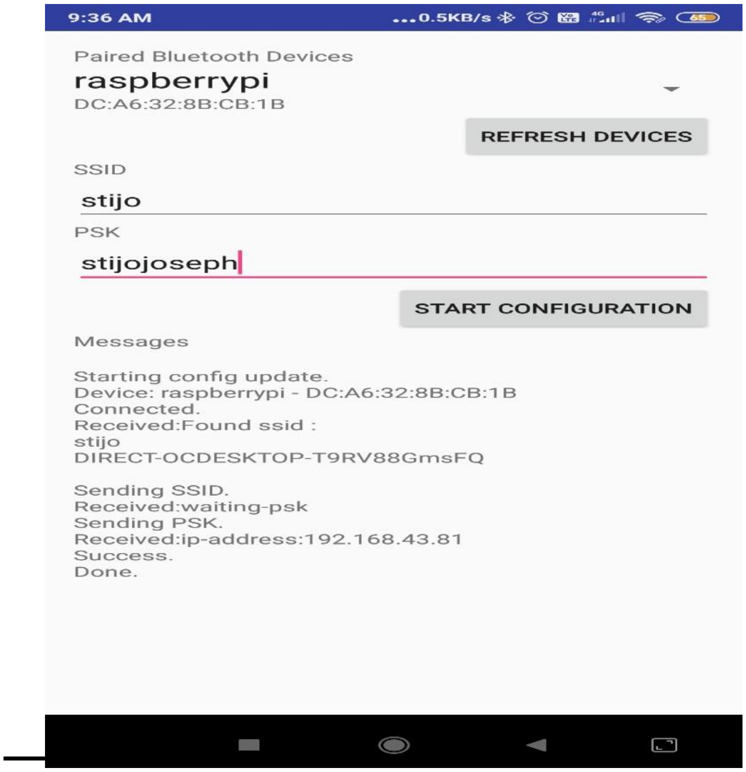
Messages

OUTPUT ON RASPI



```
pi@raspberrypi:~ $ cd bluetoothconfig
pi@raspberrypi:~/bluetoothconfig $ sudo python3 run_main.py
wifi password configurations starting
Waiting for connection on RFCOMM channel 1
Accepted connection from  ('D8:32:E3:45:4A:E1', 1)
Waiting for SSID...
ssid received
stijo
Waiting for PSK...
psk received
stijojoseph
the recieved SSID matches with the discovered SSID
*********wait for 10 seconds please******************
OK
```

- **AFTER WAITING FOR 10 SECONDS**
- **YOU WILL GET THE IP ADDRESS ON YOUR MOBILE**



Paired Bluetooth Devices
raspberrypi
DC:A6:32:8B:CB:1B

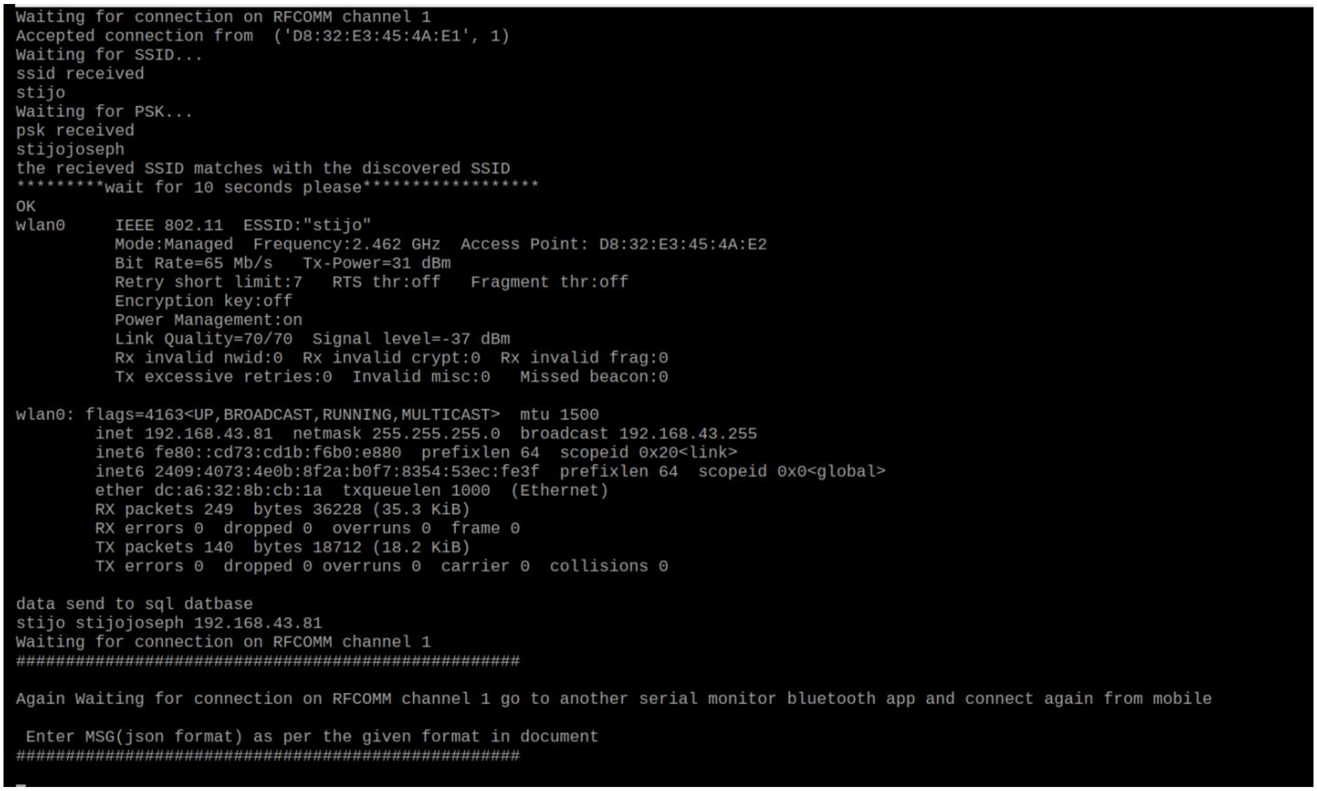
SSID
stijo

PSK
stijojoseph

Messages

Starting config update.
Device: raspberrypi - DC:A6:32:8B:CB:1B
Connected.
Received:Found ssid :
stijo
DIRECT-OCDESKTOP-T9RV88GmsFQ

Sending SSID.
Received:waiting-psk
Sending PSK.
Received:ip-address:192.168.43.81
Success.
Done.

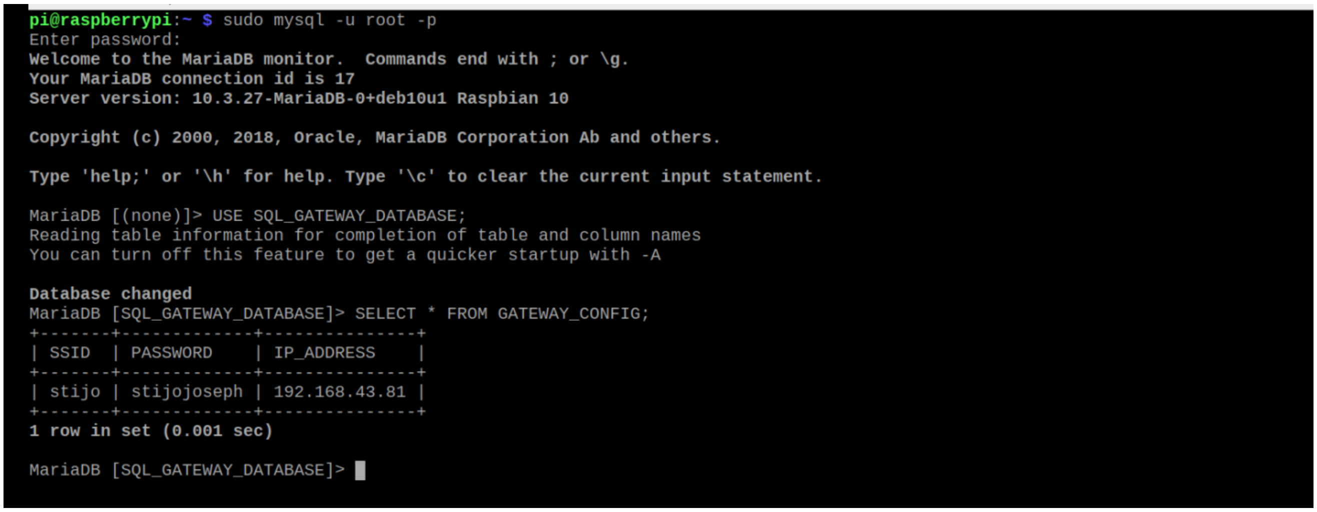


```
Waiting for connection on RFCOMM channel 1
Accepted connection from  ('D8:32:E3:45:4A:E1', 1)
Waiting for SSID...
ssid received
stijo
Waiting for PSK...
psk received
stijojoseph
the recieved SSID matches with the discovered SSID
*********wait for 10 seconds please******************
OK
wlan0     IEEE 802.11  ESSID:"stijo"
          Mode:Managed  Frequency:2.462 GHz  Access Point: D8:32:E3:45:4A:E2
          Bit Rate=65 Mb/s   Tx-Power=31 dBm
          Retry short limit:7   RTS thr:off   Fragment thr:off
          Encryption key:off
          Power Management:on
          Link Quality=70/70  Signal level=-37 dBm
          Rx invalid nwid:0  Rx invalid crypt:0  Rx invalid frag:0
          Tx excessive retries:0  Invalid misc:0   Missed beacon:0

wlan0: flags=4163<UP,BROADCAST,RUNNING,MULTICAST>  mtu 1500
        inet 192.168.43.81  netmask 255.255.255.0  broadcast 192.168.43.255
        inet6 fe80::cd73:cd1b:f6b0:e880  prefixlen 64  scopeid 0x20<link>
        inet6 2409:4073:4e0b:8f2a:b0f7:8354:53ec:fe3f  prefixlen 64  scopeid 0x0<global>
        ether dc:a6:32:8b:cb:1a  txqueuelen 1000  (Ethernet)
        RX packets 249  bytes 36228 (35.3 KiB)
        RX errors 0  dropped 0  overruns 0  frame 0
        TX packets 140  bytes 18712 (18.2 KiB)
        TX errors 0  dropped 0 overruns 0  carrier 0  collisions 0

data send to sql datbase
stijo stijojoseph 192.168.43.81
Waiting for connection on RFCOMM channel 1
###################################################

Again Waiting for connection on RFCOMM channel 1 go to another serial monitor bluetooth app and connect again from mobile

 Enter MSG(json format) as per the given format in document
###################################################
```

CONTINUATION TO THIS PROGRAM SENDING USERNAME,PASSWORD,SECURITY IN JSON FORMAT IS IN NEXT PAGE

IF YOU SEE SIMILAR KIND OF OUTPUT IT ,MEANS THE DATA IS SENT TO THE SQL DATABASE,THE IP ADDRESS,SSID ,PASSWORD ARE STORED IN THE GATEWAY_CONFIG TABLE AND THIS IPADDRESS WILL BE USED AS MQTTBROKER ADDRESS FOR SUBSCRIBING AND PUBLISHING FOR THE REST OF THE PROGRAMS

CHECK THE DATABASE FOR CONFIRMATION



```
pi@raspberrypi:~ $ sudo mysql -u root -p
Enter password:
Welcome to the MariaDB monitor.  Commands end with ; or \g.
Your MariaDB connection id is 17
Server version: 10.3.27-MariaDB-0+deb10u1 Raspbian 10

Copyright (c) 2000, 2018, Oracle, MariaDB Corporation Ab and others.

Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.

MariaDB [(none)]> USE SQL_GATEWAY_DATABASE;
Reading table information for completion of table and column names
You can turn off this feature to get a quicker startup with -A

Database changed
MariaDB [SQL_GATEWAY_DATABASE]> SELECT * FROM GATEWAY_CONFIG;
+-------+-------------+--------------+
| SSID  | PASSWORD    | IP_ADDRESS   |
+-------+-------------+--------------+
| stijo | stijojoseph | 192.168.43.81 |
+-------+-------------+--------------+
1 row in set (0.001 sec)

MariaDB [SQL_GATEWAY_DATABASE]> 
```

Set Gateway settings via WIFI server (Secure Login) – First time default fixed password.

Step 1- Now app knows the ip address of raspberry pi, user will turn off the mobile Bluetooth and switch to wi-fi. App will automatically connect with the ip of gateway.

Step 2- Now app will request for the http webpage so that user can configure the gateway for mqtt. Python running web server will response with a web page.

Step 3- Webpage will be secured with a default id and password, customer will enter the default id and password and if it is correct webpage will show a option to change the customer's id, password and security key.

After the mobile gateway setting are over ,an another Bluetooth channel will start

Here the userid ,password,securitykey,command is sent via Bluetooth itself.For this I use another Bluetooth app with Serial monitor

In the mobile app type in the serial monitor

App2 download link

https://play.google.com/store/apps/details?id=de.kai_morich.serial_bluetooth_terminal

If you want to reconfigure ssid and password after uploading this username,password,securitykey , put “wificonfig” as json value of command , Here the command value is to decide which mode the program should go into

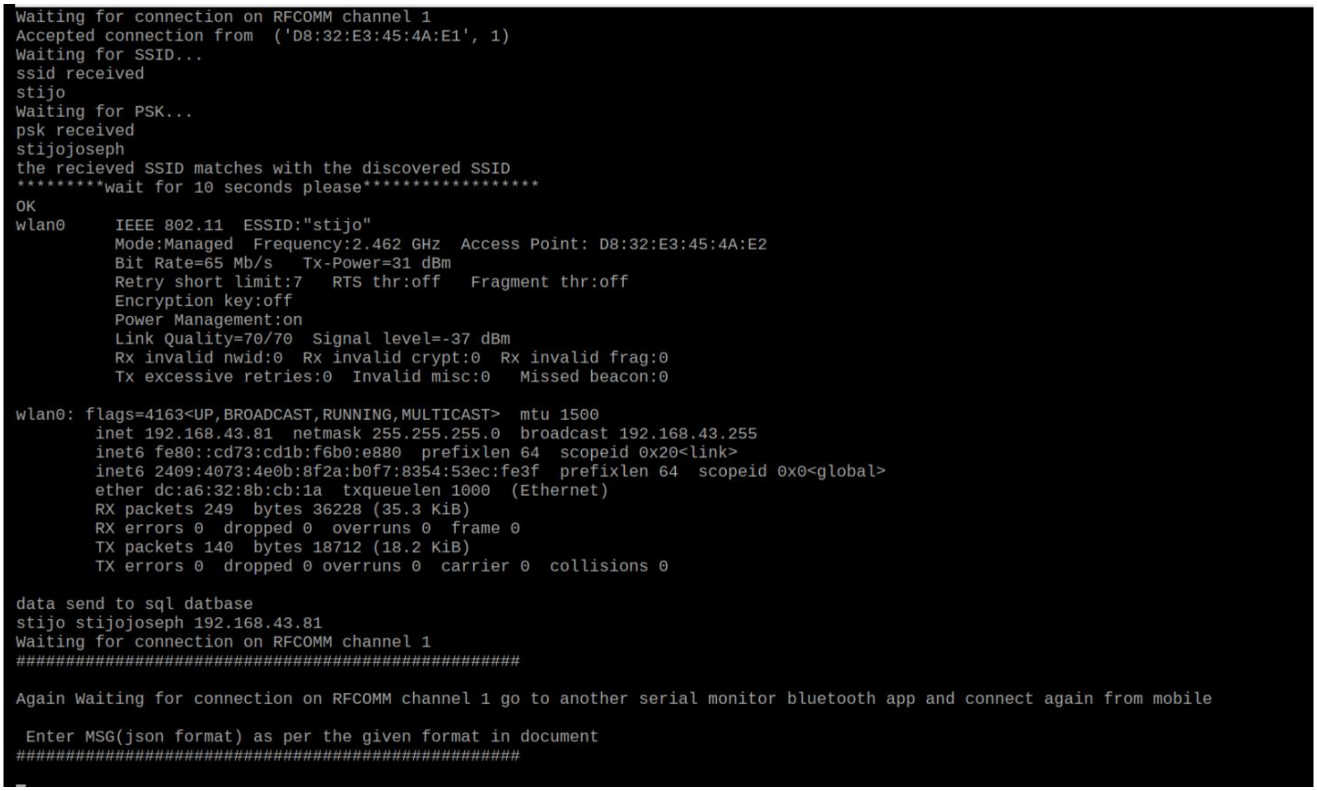
or else to add more username,password etc to sql keep the command value as “config”

Msg format:

{ “userid” : “your_username” , “password” : “your_password” , “securitykey” : “secure”, “command” : “wificonfig”}

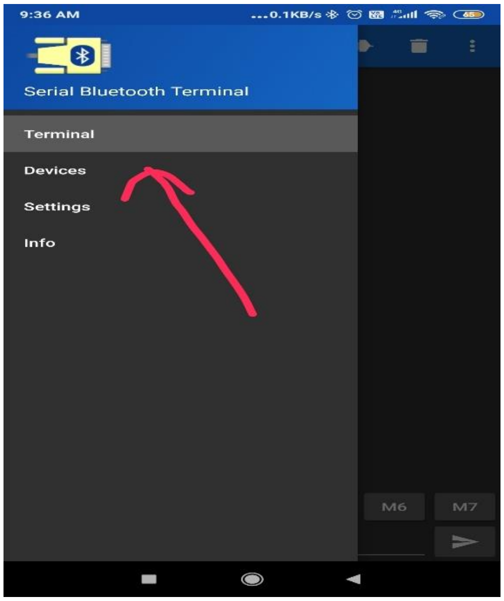
//program restarts to configure ssid password and all afer saving this uername password etc

PREVIOUS PROGRAM IN RUNNING MODE

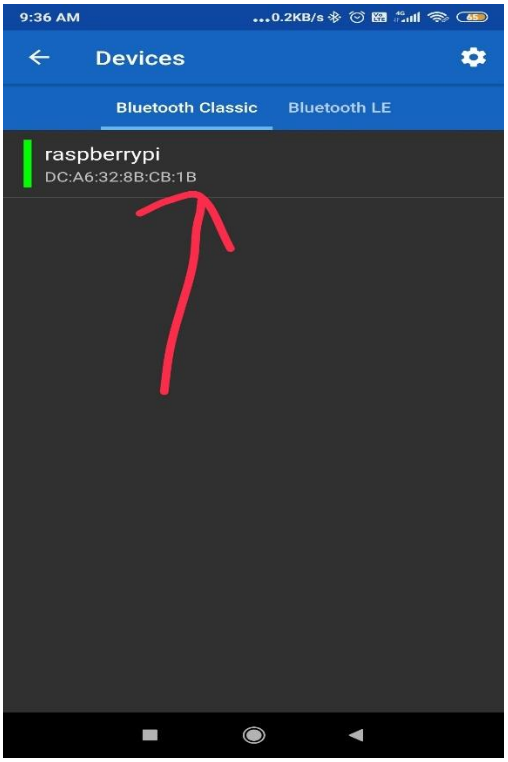


```
Waiting for connection on RFCOMM channel 1
Accepted connection from  ('D8:32:E3:45:4A:E1', 1)
Waiting for SSID...
ssid received
stijo
Waiting for PSK...
psk received
stijojoseph
the recieved SSID matches with the discovered SSID
*********wait for 10 seconds please*****************
OK
wlan0     IEEE 802.11  ESSID:"stijo"
          Mode:Managed  Frequency:2.462 GHz  Access Point: D8:32:E3:45:4A:E2
          Bit Rate=65 Mb/s   Tx-Power=31 dBm
          Retry short limit:7   RTS thr:off   Fragment thr:off
          Encryption key:off
          Power Management:on
          Link Quality=70/70  Signal level=-37 dBm
          Rx invalid nwid:0  Rx invalid crypt:0  Rx invalid frag:0
          Tx excessive retries:0  Invalid misc:0   Missed beacon:0

wlan0: flags=4163<UP,BROADCAST,RUNNING,MULTICAST>  mtu 1500
        inet 192.168.43.81  netmask 255.255.255.0  broadcast 192.168.43.255
        inet6 fe80::cd73:cd1b:f6b0:e880  prefixlen 64  scopeid 0x20<link>
        inet6 2409:4073:4e0b:8f2a:b0f7:8354:53ec:fe3f  prefixlen 64  scopeid 0x0<global>
        ether dc:a6:32:8b:cb:1a  txqueuelen 1000  (Ethernet)
        RX packets 249  bytes 36228 (35.3 KiB)
        RX errors 0  dropped 0  overruns 0  frame 0
        TX packets 140  bytes 18712 (18.2 KiB)
        TX errors 0  dropped 0 overruns 0  carrier 0  collisions 0

data send to sql datbase
stijo stijojoseph 192.168.43.81
Waiting for connection on RFCOMM channel 1
###################################################

Again Waiting for connection on RFCOMM channel 1 go to another serial monitor bluetooth app and connect again from mobile

 Enter MSG(json format) as per the given format in document
###################################################
```

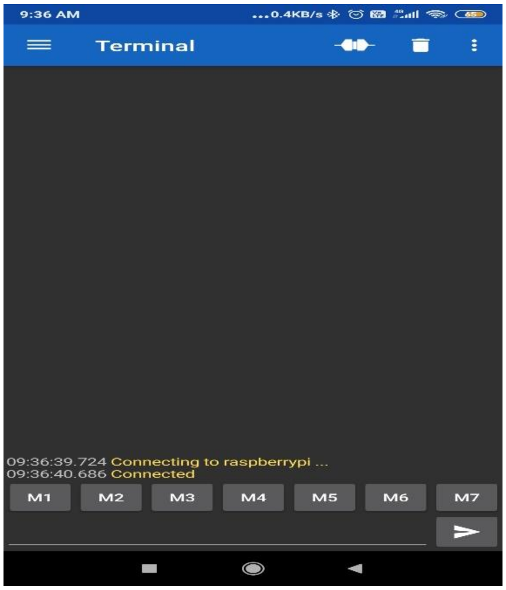
TOUCH ON DEVICES



SELECT RASPBERRYPI



AFTER SELECTING RASPBERRY PI YOU CAN SEE CONNECTED STATUS ON MOBILE TERMINAL



```
09:36:39.724 Connecting to raspberrypi ...
09:36:40.686 Connected
```

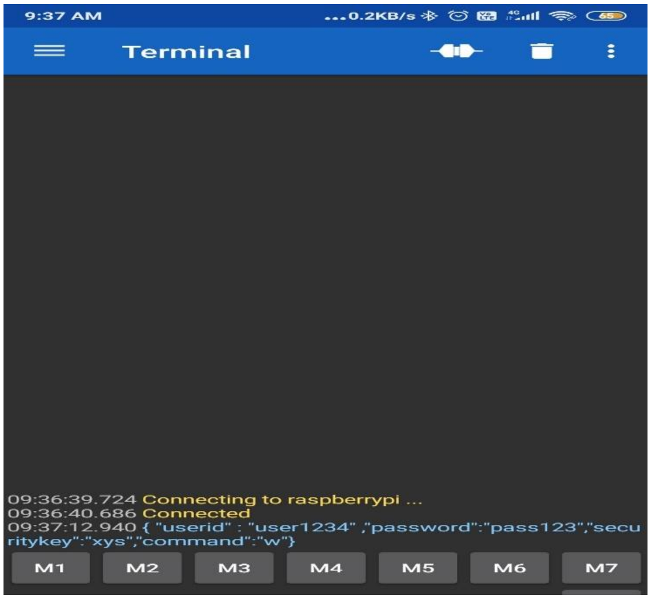
Msg format:

{ “userid” : “your_username” , “password” : “your_password” , “securitykey” : “secure”, “command” : “config”}

In the command value if anyother value other than “wificonfig” is given the program will move into user details addition mode ,for wifi settings configure again u may add “wificonfig” as command value else its your wish

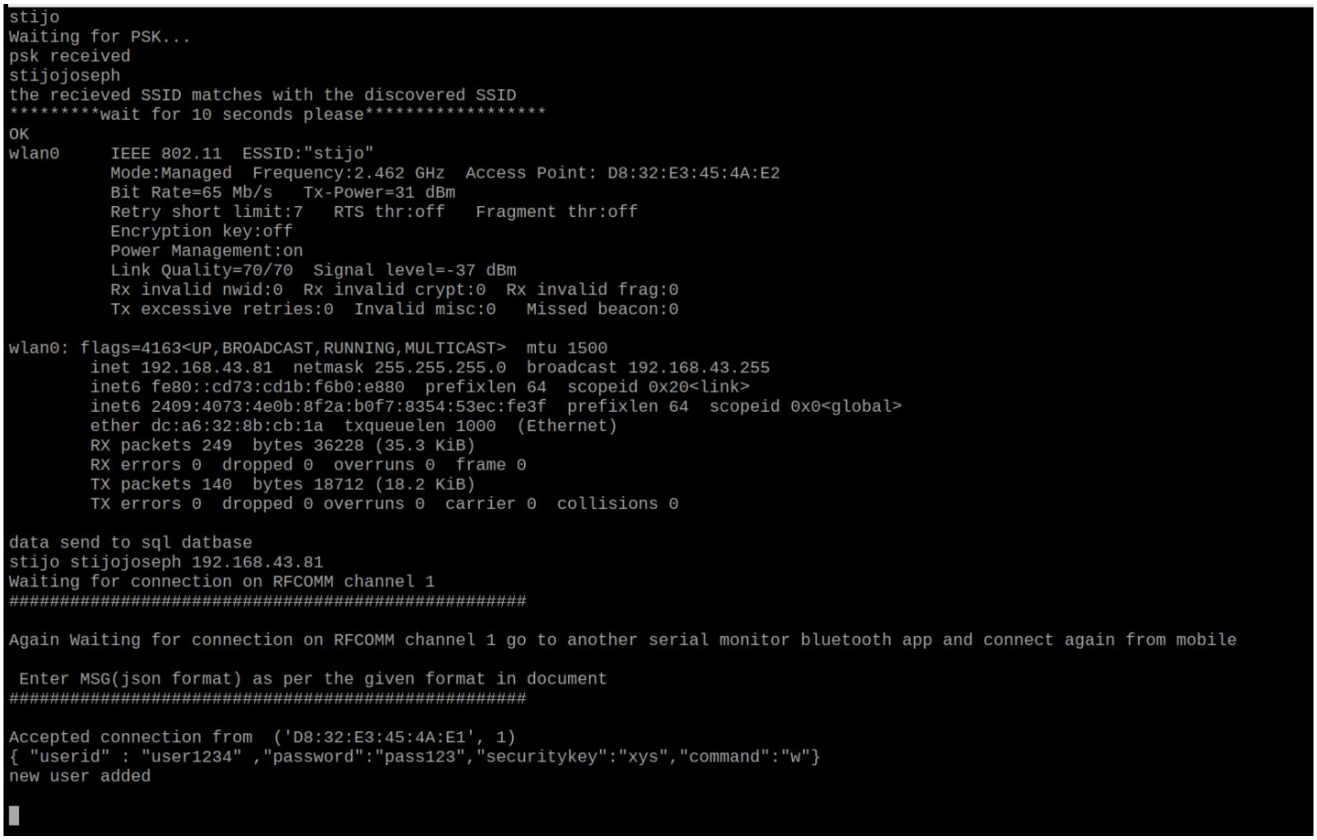
And touch send option

After sending the terminal will look like this



```
09:36:39.724 Connecting to raspberrypi ...
09:36:40.686 Connected
09:37:12.940 { "userid" : "user1234" ,"password":"pass123","secu
ritykey":"xys","command":"w"}
```

Pi terminal will be like this



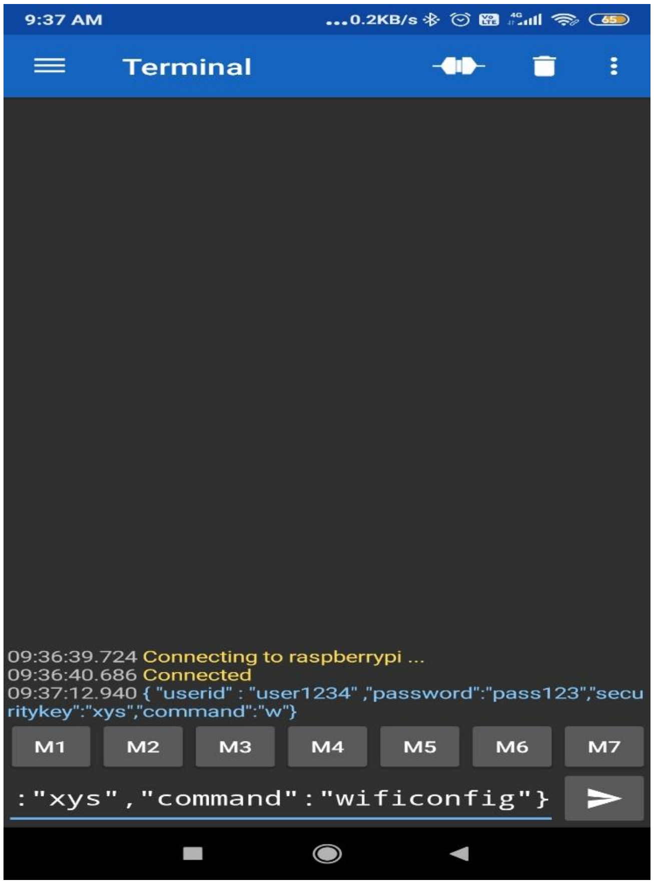
```
stijo
Waiting for PSK...
psk received
stijojoseph
the recieved SSID matches with the discovered SSID
*********wait for 10 seconds please******************
OK
wlan0     IEEE 802.11  ESSID:"stijo"
          Mode:Managed  Frequency:2.462 GHz  Access Point: D8:32:E3:45:4A:E2
          Bit Rate=65 Mb/s   Tx-Power=31 dBm
          Retry short limit:7   RTS thr:off   Fragment thr:off
          Encryption key:off
          Power Management:on
          Link Quality=70/70  Signal level=-37 dBm
          Rx invalid nwid:0  Rx invalid crypt:0  Rx invalid frag:0
          Tx excessive retries:0  Invalid misc:0   Missed beacon:0

wlan0: flags=4163<UP,BROADCAST,RUNNING,MULTICAST>  mtu 1500
        inet 192.168.43.81  netmask 255.255.255.0  broadcast 192.168.43.255
        inet6 fe80::cd73:cd1b:f6b0:e880  prefixlen 64  scopeid 0x20<link>
        inet6 2409:4073:4e0b:8f2a:b0f7:8354:53ec:fe3f  prefixlen 64  scopeid 0x0<global>
        ether dc:a6:32:8b:cb:1a  txqueuelen 1000  (Ethernet)
        RX packets 249  bytes 36228 (35.3 KiB)
        RX errors 0  dropped 0  overruns 0  frame 0
        TX packets 140  bytes 18712 (18.2 KiB)
        TX errors 0  dropped 0 overruns 0  carrier 0  collisions 0

data send to sql datbase
stijo stijojoseph 192.168.43.81
Waiting for connection on RFCOMM channel 1
###################################################

Again Waiting for connection on RFCOMM channel 1 go to another serial monitor bluetooth app and connect again from mobile

 Enter MSG(json format) as per the given format in document
###################################################

Accepted connection from  ('D8:32:E3:45:4A:E1', 1)
{ "userid" : "user1234" ,"password":"pass123","securitykey":"xys","command":"w"}
new user added

█
```

//program restarts to add more username ,password,securitykey

Msg format

{ “userid” : “your_username” , “password” : “your_password” , “securitykey” : “secure”, “command” : “wificonfig”}

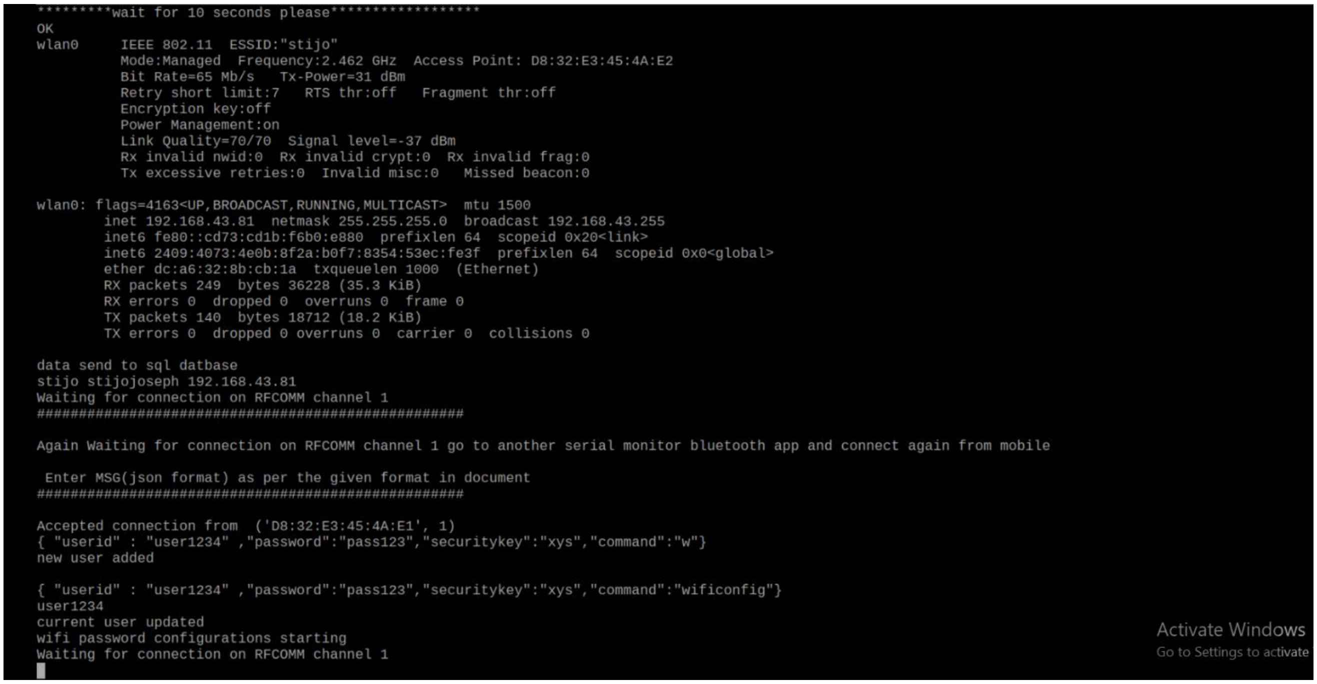
this msgformat is for adding user details and shift to wifi configuration mode



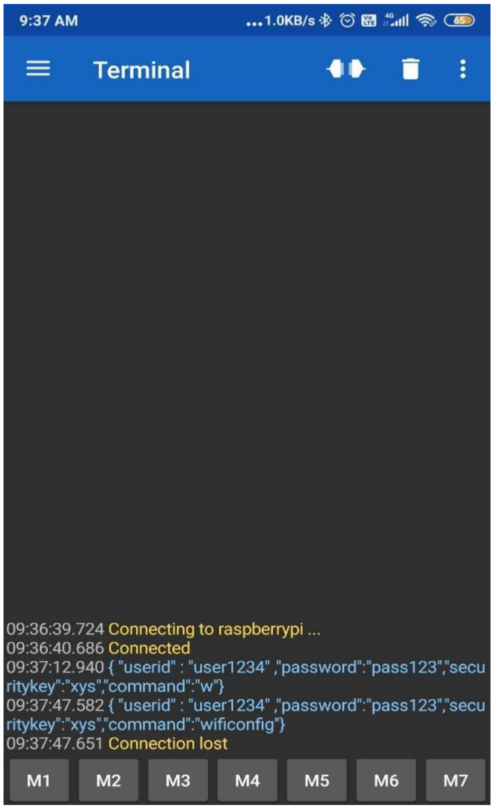
```
09:36:39.724 Connecting to raspberrypi ...
09:36:40.686 Connected
09:37:12.940 { "userid" : "user1234" ,"password":"pass123","secu
ritykey":"xys","command":"w"}
```

```
:"xys","command":"wificonfig"}
```

After sending



```
*********wait for 10 seconds please******************
OK
wlan0     IEEE 802.11  ESSID:"stijo"
          Mode:Managed  Frequency:2.462 GHz  Access Point: D8:32:E3:45:4A:E2
          Bit Rate=65 Mb/s   Tx-Power=31 dBm
          Retry short limit:7   RTS thr:off   Fragment thr:off
          Encryption key:off
          Power Management:on
          Link Quality=70/70  Signal level=-37 dBm
          Rx invalid nwid:0  Rx invalid crypt:0  Rx invalid frag:0
          Tx excessive retries:0  Invalid misc:0   Missed beacon:0

wlan0: flags=4163<UP,BROADCAST,RUNNING,MULTICAST>  mtu 1500
        inet 192.168.43.81  netmask 255.255.255.0  broadcast 192.168.43.255
        inet6 fe80::cd73:cd1b:f6b0:e880  prefixlen 64  scopeid 0x20<link>
        inet6 2409:4073:4e0b:8f2a:b0f7:8354:53ec:fe3f  prefixlen 64  scopeid 0x0<global>
        ether dc:a6:32:8b:cb:1a  txqueuelen 1000  (Ethernet)
        RX packets 249  bytes 36228 (35.3 KiB)
        RX errors 0  dropped 0  overruns 0  frame 0
        TX packets 140  bytes 18712 (18.2 KiB)
        TX errors 0  dropped 0 overruns 0  carrier 0  collisions 0

data send to sql datbase
stijo stijojoseph 192.168.43.81
Waiting for connection on RFCOMM channel 1
###################################################

Again Waiting for connection on RFCOMM channel 1 go to another serial monitor bluetooth app and connect again from mobile

 Enter MSG(json format) as per the given format in document
###################################################

Accepted connection from  ('D8:32:E3:45:4A:E1', 1)
{ "userid" : "user1234" ,"password":"pass123","securitykey":"xys","command":"w"}
new user added

{ "userid" : "user1234" ,"password":"pass123","securitykey":"xys","command":"wificonfig"}
user1234
current user updated
wifi password configurations starting
Waiting for connection on RFCOMM channel 1
```



```
09:36:39.724 Connecting to raspberrypi ...
09:36:40.686 Connected
09:37:12.940 { "userid" : "user1234" ,"password":"pass123","secu
ritykey":"xys","command":"w"}
09:37:47.582 { "userid" : "user1234" ,"password":"pass123","secu
ritykey":"xys","command":"wificonfig"}
09:37:47.651 Connection lost
```

- **User details will be stored to database and connection will be lost because Bluetooth channel get restarted ,u need to use the other app for configuring wifi**

- **PRACTICALLY THIS PROGRAM SHOULD KEEP ON RUNNING FOR EVER IN BACKGROUND,MEANS ON BOOTING OF PI THIS run_main.py SHOULD BE AUTOMATICALLY EXECUTED AND THE BLUETOOTH CHANNEL WILL BE OPEN EITHER FOR WIFICONFIGURATIONS OR FOR USER DETAILS ADDING,ANY NUMBER OF USERS CAN BE ADDED**

Step 4- the customer’s id and password will be saved as admin id and password in MySQL database under a table name: users_data

User id	Password	Security key
Admin id	Xyz	abcd

Step 5- Now web server will response with a webpage to configure the following blocks-

Cloud

Server

URL End

client id

Cloud

Server

Topic

Location

Local/Both

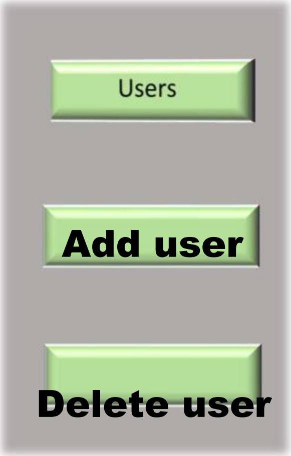
(Local + Cloud)

Device Name

Access Key

Step 6- The web server will save these credential in the MySQL with table name gateway_configuration.

Step 7- If admin user want to add more user, User will click on gateway setting in the app, again click on users in the option, gateway will response with a webpage with the following option:



Step 8- Now user will click on “Add user” option, again a window will open to enter username and password and security key and save it.

Step 9- The entered will be stored in the table in user_data in MySQL database running on gateway.

** The above steps are used to configure the gateway**

3.MQTT SETUP

VIDEO LINK

https://drive.google.com/file/d/1RFYoZWvM_MEFN_t7rnT4lyY4zpa3AOyt/view?usp=sharing

❖ Mqtt local setup

- **ON RASPBERRY PI TERMINAL**

- **Set an MQTT username and password**
- **Run this command to create an MQTT username and password (substitute YOUR-NEWMQTT-USERNAME):**

```
sudo mosquitto_passwd -c /etc/mosquitto/pwfil YOUR-NEW-MQTT-USERNAME
```

- **Now when you press Enter it will ask to create password.**

.

```
sudo nano /etc/mosquitto/mosquitto.conf
```

- **OPENING THIS WILL GIVE U A FILE, INSIDE CHANGE “pwfile” to “pwfil” .dont change or delete any other lines**

```
pid_file /var/run/mosquitto.pid
persistence true
persistence_location /var/lib/mosquitto/
log_dest file /var/log/mosquitto/mosquitto.log
allow_anonymous false
password_file /etc/mosquitto/pwfil
```

click Ctrl+O > Enter > Ctrl+X to save the config file.

Test the broker
Substitute YOUR-MQTT-USERNAME and YOUR-MQTT-PASSWORD with what you defined in
the previous step

```
mosquitto_sub -d -u YOUR-MQTT-USERNAME -P YOUR-MQTT-PASSWORD -t dev/test
```

- **You should get back a response similar to this:**

```
Client mosqsub|730-hostname sending CONNECT
Client mosqsub|730-hostname received CONNACK (0)
Client mosqsub|730-hostname sending SUBSCRIBE (Mid: 1, Topic: dev/test, QoS: 0)
Client mosqsub|730-hostname received SUBACK
```

Press Ctrl-C to break out of it.

- **. Setting**

SETTING UP THE CLOUD

Setting cloud mqtt broker
Visit https://myqtthub.com/en
□ Enter your e-mail address > username (must be unique) > Create a password
□ Now click on "Sign up with open plan".
□ Now it will send a verification mail to your registered email, go to your email and verify it. Verify email on the same PC/Laptop/pi where you have signed up for cloud broker.
□ Now again visit https://myqtthub.com/en and login using your credential then
go to home page.
□ Click on Myqtthub (Top in the left), it will open a box, click on domains
□ It will open a page MyQtt Domains, a domain name hub-username will be available there. Click on the domain name (hub-username).
□ Now click on "Bootstrap".
□ Now click on "Add bootstrap Auth credentials", create a username and password and save it. (Note it down, it will be used for client to connect with myqtt hub).

. Setting mqtt bridge to connect local mqtt broker and cloud mqtt broker

□ Before setting the mqtt bridge we have to stop running mosquitto broker-
□ Open command prompt in raspberry pi and run the command

```
sudo service mosquitto stop
```

□ Now we will open mosquitto configure file to configure it as a mqtt bridge

```
sudo nano /etc/mosquitto/mosquitto.conf
```

□ Configure this file as given below. Most of the lines will be
(Line started with '#' is comment). The words which are
written in Red colour, have to be modified it accordingly.

```
# Place your local configuration in /etc/mosquitto/conf.d/
# A full description of the configuration file is at
# /usr/share/doc/mosquitto/examples/mosquitto.conf.example



pid_file /var/run/mosquitto.pid
persistence true
persistence_location /var/lib/mosquitto/
log_dest file /var/log/mosquitto/mosquitto.log
allow_anonymous false
password_file /etc/mosquitto/pwfiles
#port for local mqtt broker
listener 1884
# connection name
connection node02
# address and port of broker with which you want to create a bridge. Here
#myqtthub
address node02.myqtthub.com:1883
cleansession true
keepalive_interval 60
notifications false
# start automatically after 30 sec. when pi starts.
start_type automatic
try_private true
# create a client id for the bridge
remote_clientid clientid
# enter password which you have created in bootstrap in myqtthub
remote_password password  #put your boot strap cred password here ,from cloud
# enter username which you have created in bootstrap in myqtthub
remote_username username #put your boot strap cred username here, from cloud
# subscribe and publish all topic both sides.
```

topic # both
click Ctrl+O > Enter > Ctrl+X to save the config file.
Now reboot the pi (sudo reboot)

 After rebooting open command prompt and enter

```
sudo mosquitto -c mosquitto.conf -v
```

 Open another command prompt and run

```
mosquitto -v
```

Now mosquitto is running.

Download app

https://play.google.com/store/apps/details?id=snr.lab.iotmqttpanel.prod

3.Mobile - Control Box

Set Control Box configuration – via control box hotspot connection (secure login)

Basic idea:

Inorder to make the esp8266/controlbox to Send the ssid,password,ipaddress, appliances names,Room name etc and rest all the configuration details we are going to send esp8266

By making it as a web server.

Means esp8266 upon uploading the code act as webserver

That is it starts a hotspot ,connect to the hotspot with mobile /pc

You need to configure the details in the webpage, and up on submittion the datas .your details will be saved in its EEPROM

Config file.Next time restarting it gets connected to the wifi

THE PROCEDURE PERFORMED BY USER

Step 1- User will plug and turn ON the control box.

Step 2- Control box will open its hotspot with a default Wi-Fi SSID and password.

Step 3- Customer will connect the smartphone wi-fi with the hotspot of control box and open the app and choose option to configure the control box.

Step 4- App will connect with control box with a default ip address.

Step 5- Control box will response a webpage to configure it.

Step 6- Now customer/app will configure the following blocks and save it –

Control Box Name

Room name (where it is to be

implemented) Access Key

Location

Appliance 1

Appliance 2 (so on till Appliance 6, if

available) PWM Appliance 1

PWM Appliance

2 Gateway ip,

port Wi-Fi SSID

Wi-Fi pass

Step 7- The above configuration will also be saved in app and when app again connect with gateway it will sync the control box configuration with gateway using a webpage. Gateway will store the control box information in a table name control_box_configuration.

VIDEO LINK-

https://drive.google.com/file/d/1Tgxda4HJ7a4wMc5zadwMRV1wgKHXAKHi/view?usp=sharing

PROCEDURE NEEDED TO BE PERFORMED BY THE PROGRAMMER TO SETUP THE CONTROL BOX

1.Connect the esp8266/controlbox to pc ,open the Arduino ide

2.Compile the code ,before uploading the connect “D0” pin of esp8266 to its 3V pin with jumper wire

There are two modes for this esp8266

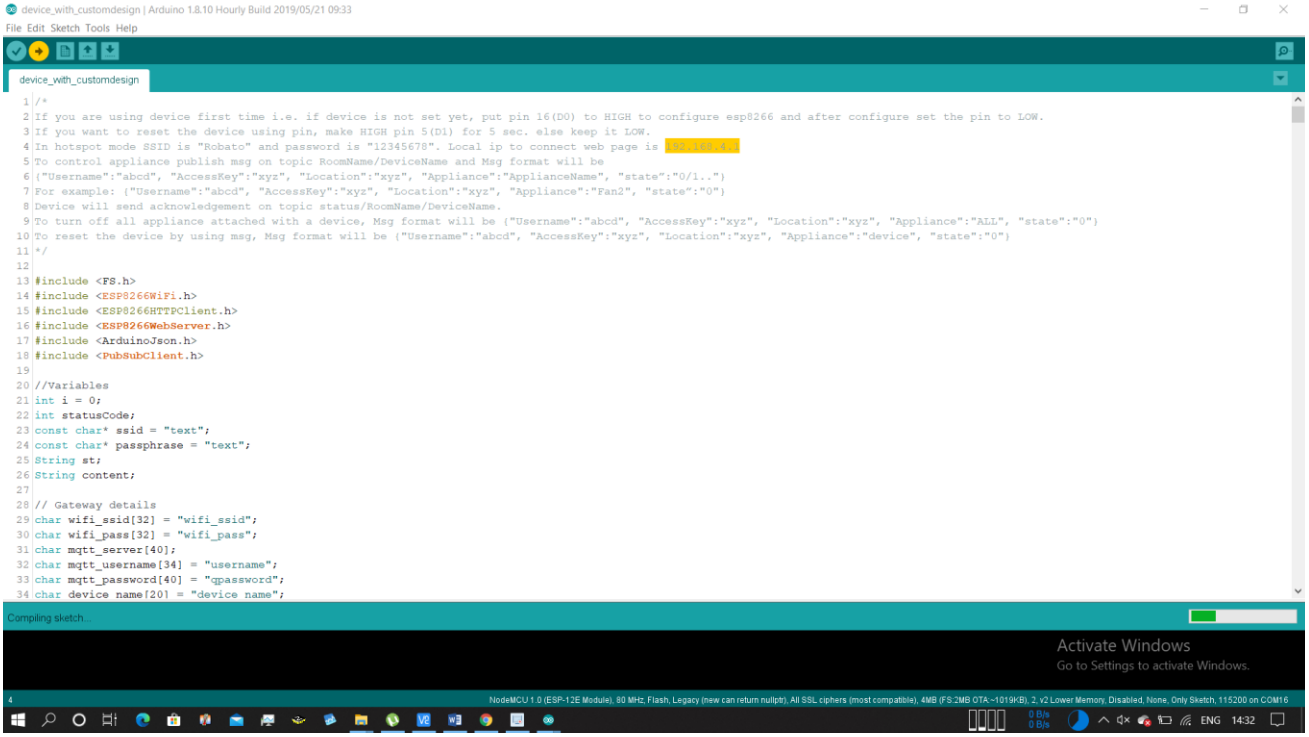
a. Web server – to configure the details to work as control box

b. Control box – for appliance on/off and sending ack to gateway via mqtt

note:by connect D0 to 3V when are making it as webserver to configure details here

upload the code and open the serial monitor to see the output

D0 should be connected to 3v



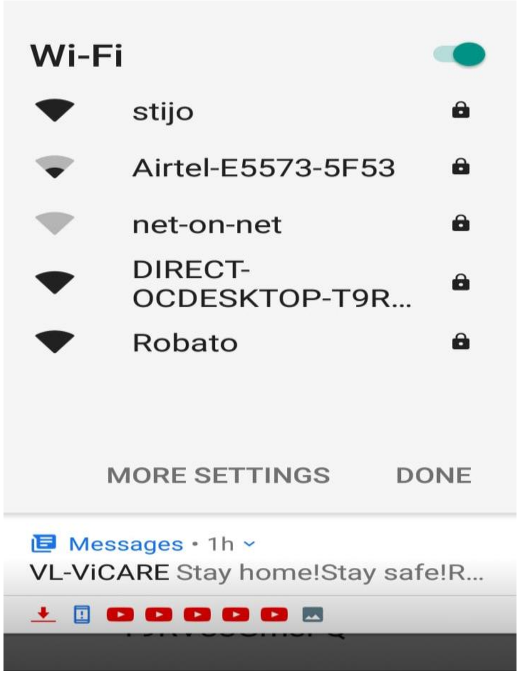
device_with_customdesign | Arduino 1.8.10 Hourly Build 2019/05/21 09:33

```
/*
If you are using device first time i.e. if device is not set yet, put pin 16(D0) to HIGH to configure esp8266 and after configure set the pin to LOW.
If you want to reset the device using pin, make HIGH pin 5(D1) for 5 sec. else keep it LOW.
In hotspot mode SSID is "Robato" and password is "12345678". Local ip to connect web page is 192.168.4.1
To control appliance publish msg on topic RoomName/DeviceName and Msg format will be
{"Username":"abcd", "AccessKey":"xyz", "Location":"xyz", "Appliance":"ApplianceName", "state":"0/1.."}
For example: {"Username":"abcd", "AccessKey":"xyz", "Location":"xyz", "Appliance":"Fan2", "state":"0"}
Device will send acknowledgement on topic status/RoomName/DeviceName.
To turn off all appliance attached with a device, Msg format will be {"Username":"abcd", "AccessKey":"xyz", "Location":"xyz", "Appliance":"ALL", "state":"0"}
To reset the device by using msg, Msg format will be {"Username":"abcd", "AccessKey":"xyz", "Location":"xyz", "Appliance":"device", "state":"0"}
*/

#include <FS.h>
#include <ESP8266WiFi.h>
#include <ESP8266HTTPClient.h>
#include <ESP8266WebServer.h>
#include <ArduinoJson.h>
#include <PubSubClient.h>

//Variables
int i = 0;
int statusCode;
const char* ssid = "text";
const char* passphrase = "text";
String st;
String content;

// Gateway details
char wifi_ssid[32] = "wifi_ssid";
char wifi_pass[32] = "wifi_pass";
char mqtt_server[40];
char mqtt_username[34] = "username";
char mqtt_password[40] = "qpassword";
char device_name[20] = "device_name";
```

```
Compiling sketch...
```

NodeMCU 1.0 (ESP-12E Module), 80 MHz, Flash, Legacy (new can return nullptr), All SSL ciphers (most compatible), 4MB (FS:2MB OTA:~1019KB), 2, v2 Lower Memory, Disabled, None, Only Sketch, 115200 on COM16

4.Switch ON any mobile or pc wifi and search for “Robato” hotspot

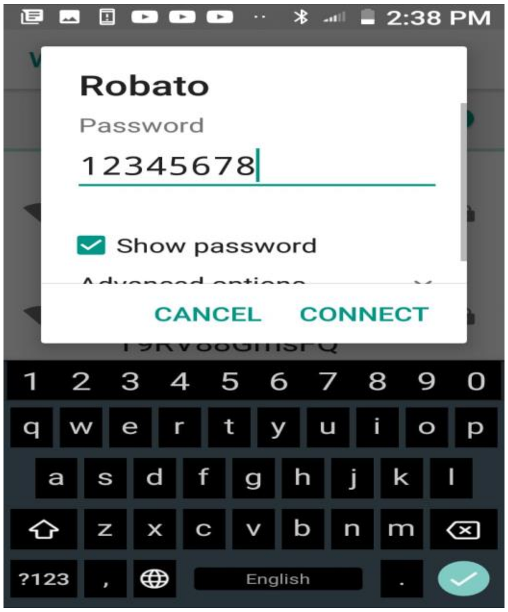


Wi-Fi

- stijo
- Airtel-E5573-5F53
- net-on-net
- DIRECT-OCDESKTOP-T9R...
- Robato

Password is

12345678



Robato

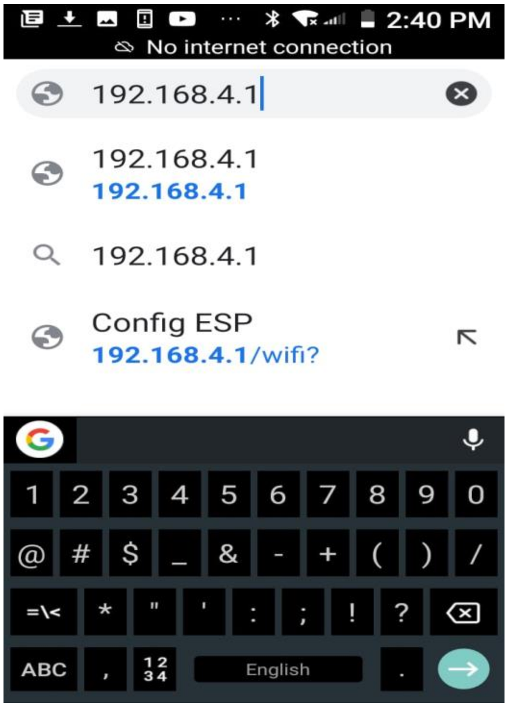
Password: 12345678

☑ Show password

5.After connecting the Robato Hotspot in the same mobile open

Any web browser like chrome type this local ip in the search bar

192.168.4.1


2:40 PM

No internet connection

192.168.4.1

192.168.4.1
192.168.4.1

192.168.4.1

Config ESP
192.168.4.1/wifi?

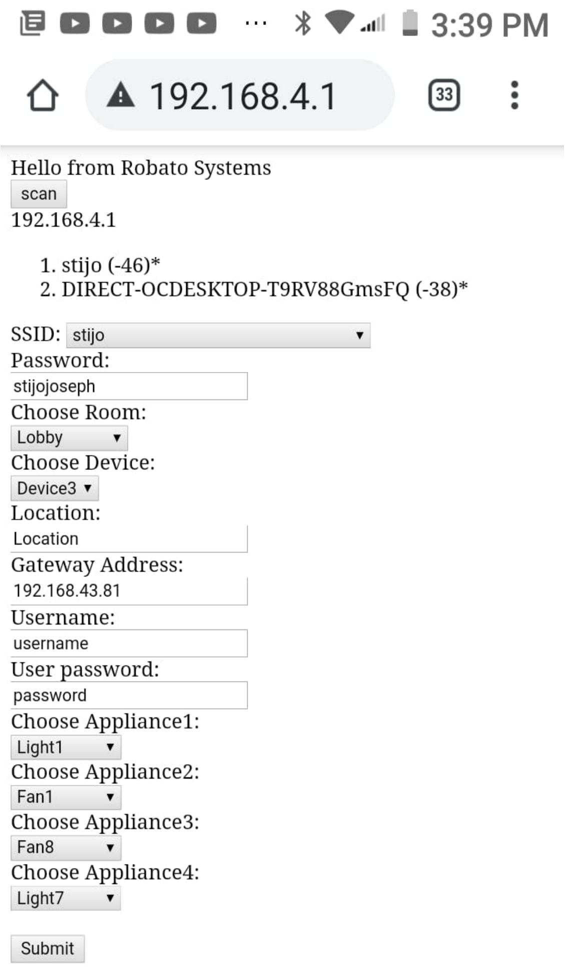
Configure the everything in the webpage,

Type the ip address you got from the mobile Bluetooth in the gatway address coloumn,

Username

Password

These coloumns should be filled accordingly by which you have configured in local mqtt username and password



Hello from Robato Systems

192.168.4.1

1. stijo (-46)*
2. DIRECT-OCDESKTOP-T9RV88GmsFQ (-38)*

SSID: stijo ▾

Password:
stijojoseph

Choose Room:
Lobby ▾

Choose Device:
Device3 ▾

Location:
Location

Gateway Address:
192.168.43.81

Username:
username

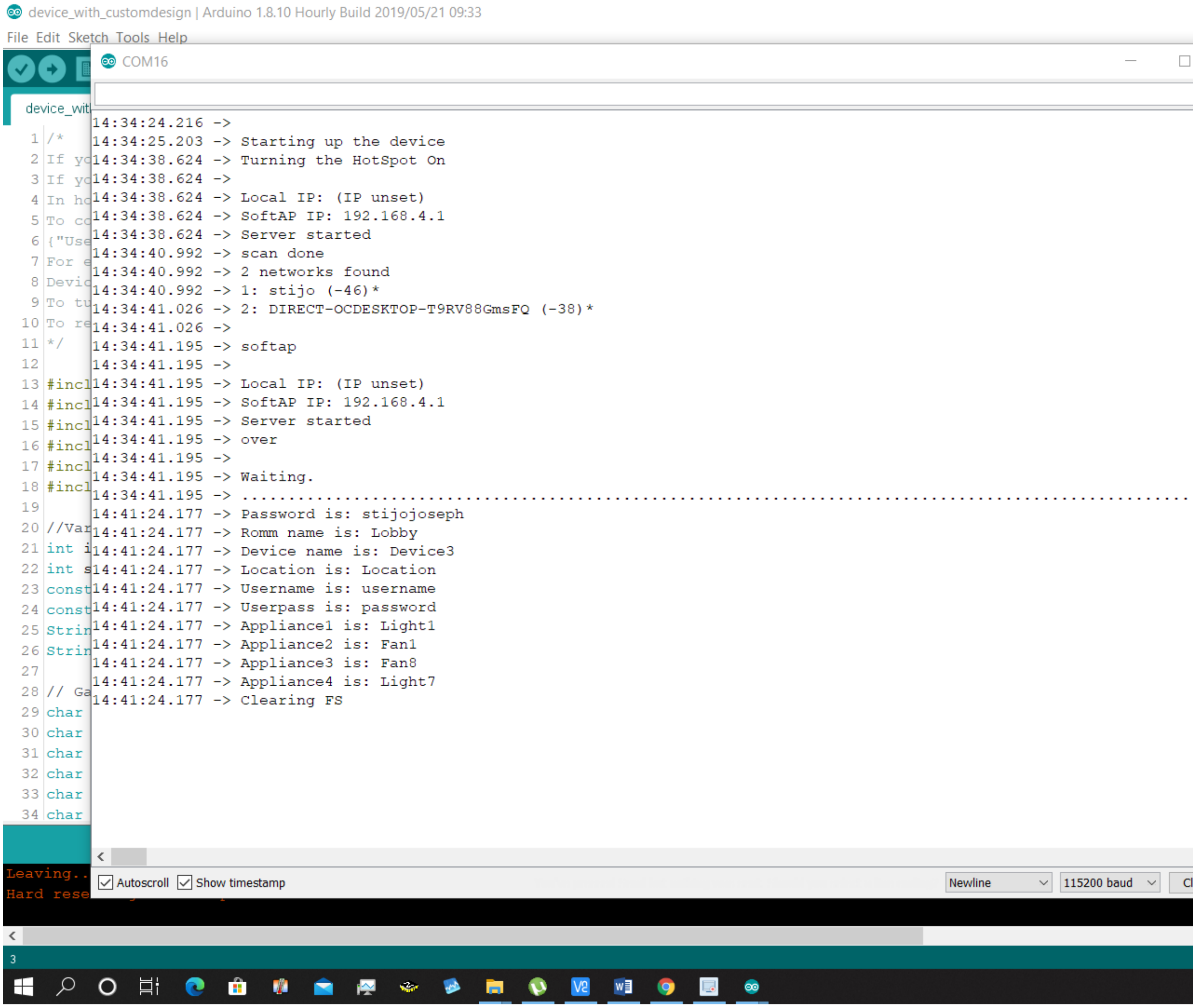
User password:
password

Choose Appliance1:
Light1 ▾

Choose Appliance2:
Fan1 ▾

Choose Appliance3:
Fan8 ▾

Choose Appliance4:
Light7 ▾


device_with_customdesign | Arduino 1.8.10 Hourly Build 2019/05/21 09:33
File Edit Sketch Tools Help
COM16
device_wit
Leaving...
Hard rese
Autoscroll
Show timestamp
Newline
115200 baud

```
14:34:24.216 ->
14:34:25.203 -> Starting up the device
14:34:38.624 -> Turning the HotSpot On
14:34:38.624 ->
14:34:38.624 -> Local IP: (IP unset)
14:34:38.624 -> SoftAP IP: 192.168.4.1
14:34:38.624 -> Server started
14:34:40.992 -> scan done
14:34:40.992 -> 2 networks found
14:34:40.992 -> 1: stijo (-46)*
14:34:41.026 -> 2: DIRECT-OCDESKTOP-T9RV88GmsFQ (-38)*
14:34:41.026 ->
14:34:41.195 -> softap
14:34:41.195 ->
14:34:41.195 -> Local IP: (IP unset)
14:34:41.195 -> SoftAP IP: 192.168.4.1
14:34:41.195 -> Server started
14:34:41.195 -> over
14:34:41.195 ->
14:34:41.195 -> Waiting.
14:34:41.195 -> ....................................................................................................
14:41:24.177 -> Password is: stijojoseph
14:41:24.177 -> Romm name is: Lobby
14:41:24.177 -> Device name is: Device3
14:41:24.177 -> Location is: Location
14:41:24.177 -> Username is: username
14:41:24.177 -> Userpass is: password
14:41:24.177 -> Appliance1 is: Light1
14:41:24.177 -> Appliance2 is: Fan1
14:41:24.177 -> Appliance3 is: Fan8
14:41:24.177 -> Appliance4 is: Light7
14:41:24.177 -> Clearing FS
```

If you Receive the msg “config.file “ is wriiten successfully then connect D0 pin to GND in a fastway less than 6 seconds to stop the hotspot and connect to the configured wifi

4 Gateway configurations to Database

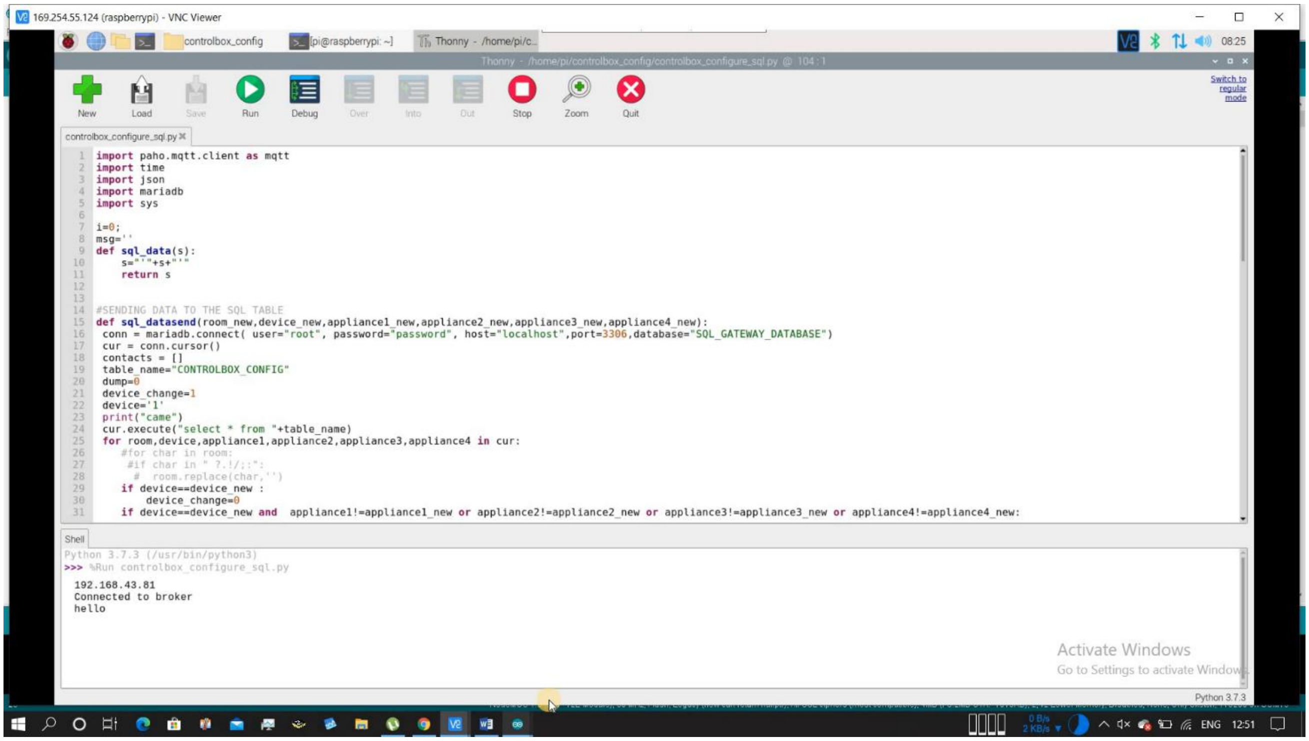
. Gateway will store the control box information in a table name control_box_configuration.

- **FOR CHECKING THAT**

OPEN THE FOLDER CONTROLBOX_CONFIG

- **RUN THE FILE IN THONNY IDE**

controlbox_configure_sql.py

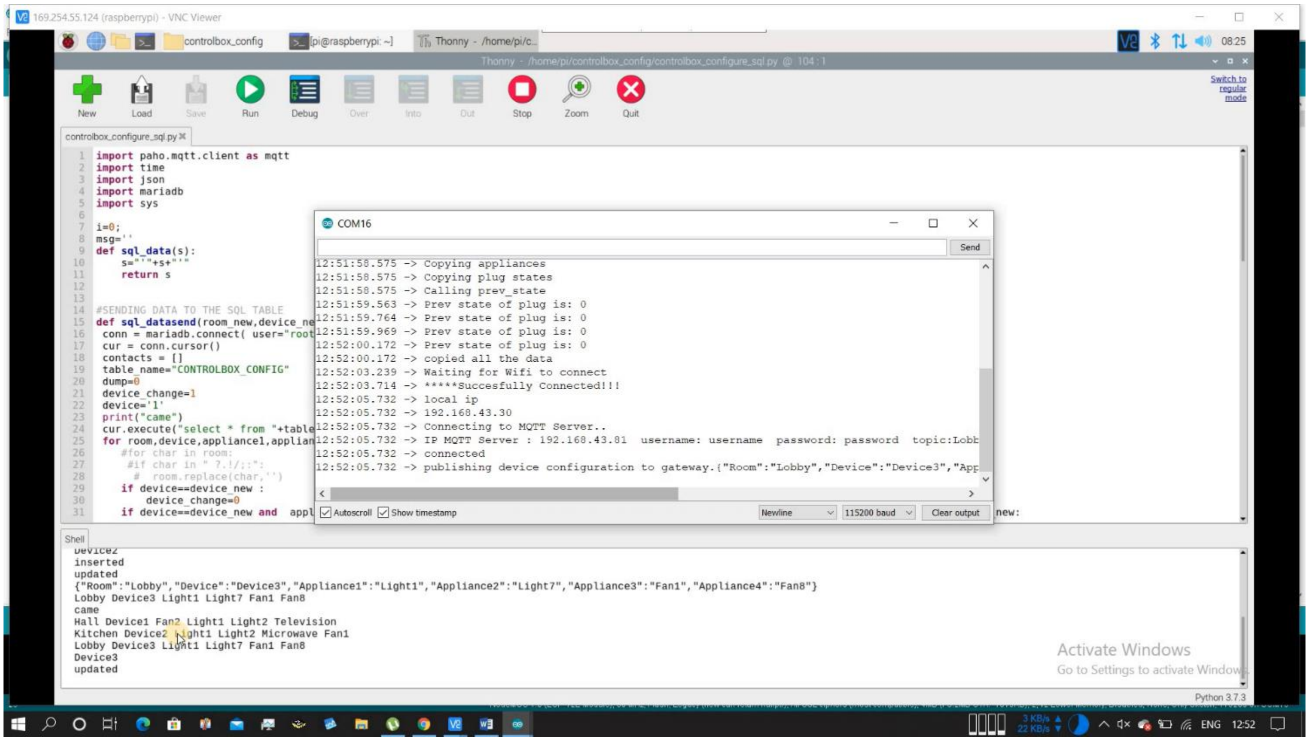

169.254.55.124 (raspberrypi) - VNC Viewer
controlbox_config
[pi@raspberrypi: ~]
Thonny - /home/pi/c...
08:25
Thonny - /home/pi/controlbox_config/controlbox_configure_sql.py @ 104 : 1
New
Load
Save
Run
Debug
Over
Into
Out
Stop
Zoom
Quit
Switch to regular mode
controlbox_configure_sql.py
Shell
Activate Windows
Go to Settings to activate Windows.
Python 3.7.3
ENG
12:51

```python
import paho.mqtt.client as mqtt
import time
import json
import mariadb
import sys

i=0;
msg=''
def sql_data(s):
    s="'"+s+"'"
    return s


#SENDING DATA TO THE SQL TABLE
def sql_datasend(room_new,device_new,appliance1_new,appliance2_new,appliance3_new,appliance4_new):
 conn = mariadb.connect( user="root", password="password", host="localhost",port=3306,database="SQL_GATEWAY_DATABASE")
 cur = conn.cursor()
 contacts = []
 table_name="CONTROLBOX_CONFIG"
 dump=0
 device_change=1
 device='1'
 print("came")
 cur.execute("select * from "+table_name)
 for room,device,appliance1,appliance2,appliance3,appliance4 in cur:
    #for char in room:
     #if char in " ?.!/;:":
      #  room.replace(char,'')
    if device==device_new :
        device_change=0
    if device==device_new and  appliance1!=appliance1_new or appliance2!=appliance2_new or appliance3!=appliance3_new or appliance4!=appliance4_new:
```

```
Python 3.7.3 (/usr/bin/python3)
>>> %Run controlbox_configure_sql.py

 192.168.43.81
 Connected to broker
 hello
```

(remember D0 pin should be connected to GND only)

next connect your esp8266 to pc and press reset



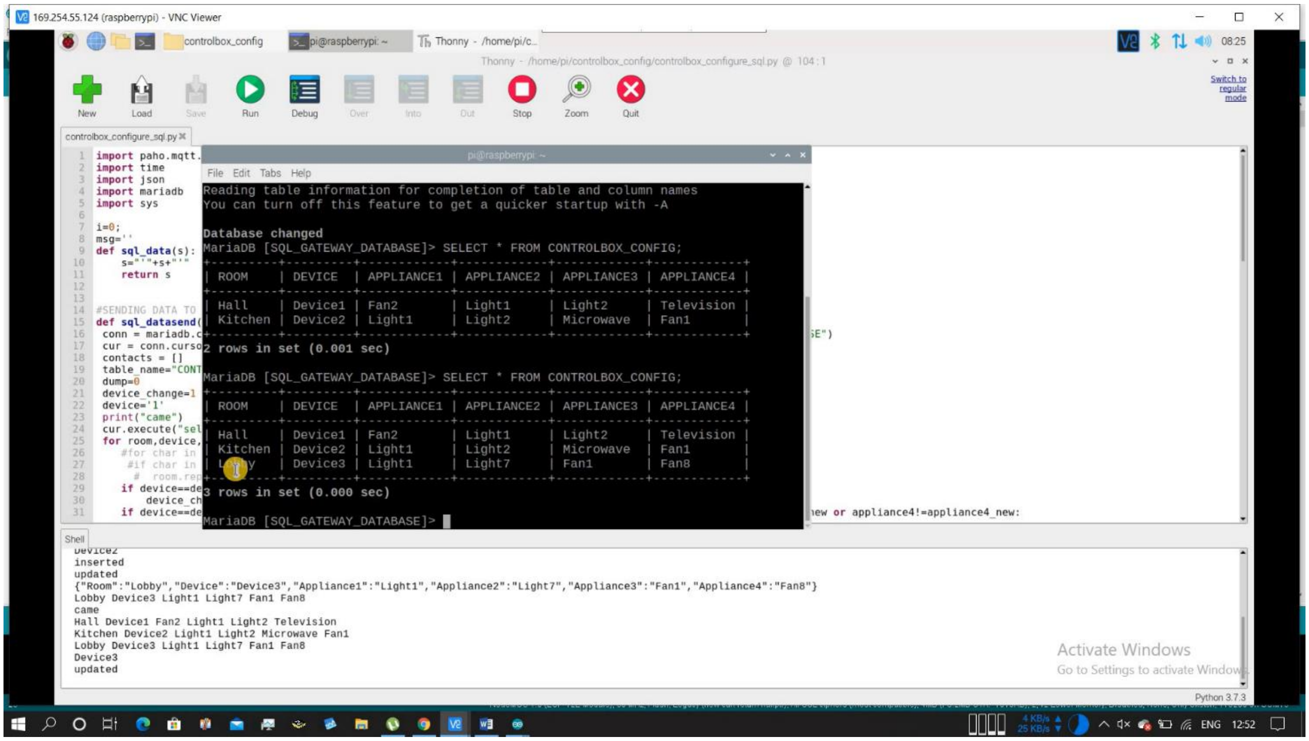
```
import paho.mqtt.client as mqtt
import time
import json
import mariadb
import sys

i=0;
msg=''
def sql_data(s):
    s="'"+s+"'"
    return s


#SENDING DATA TO THE SQL TABLE
def sql_datasend(room_new,device_ne
 conn = mariadb.connect( user="root
 cur = conn.cursor()
 contacts = []
 table_name="CONTROLBOX_CONFIG"
 dump=0
 device_change=1
 device='1'
 print("came")
 cur.execute("select * from "+table
 for room,device,appliance1,applian
    #for char in room:
     #if char in " ?.!/;:":
      #  room.replace(char,'')
    if device==device_new :
        device_change=0
    if device==device_new and  appl
```

```
12:51:58.575 -> Copying appliances
12:51:58.575 -> Copying plug states
12:51:58.575 -> Calling prev_state
12:51:59.563 -> Prev state of plug is: 0
12:51:59.764 -> Prev state of plug is: 0
12:51:59.969 -> Prev state of plug is: 0
12:52:00.172 -> Prev state of plug is: 0
12:52:00.172 -> copied all the data
12:52:03.239 -> Waiting for Wifi to connect
12:52:03.714 -> *****Succesfully Connected!!!
12:52:05.732 -> local ip
12:52:05.732 -> 192.168.43.30
12:52:05.732 -> Connecting to MQTT Server..
12:52:05.732 -> IP MQTT Server : 192.168.43.81  username: username  password: password  topic:Lobb
12:52:05.732 -> connected
12:52:05.732 -> publishing device configuration to gateway.{"Room":"Lobby","Device":"Device3","App
```

☑ Autoscroll ☑ Show timestamp Newline 115200 baud

```
inserted
updated
{"Room":"Lobby","Device":"Device3","Appliance1":"Light1","Appliance2":"Light7","Appliance3":"Fan1","Appliance4":"Fan8"}
Lobby Device3 Light1 Light7 Fan1 Fan8
came
Hall Device1 Fan2 Light1 Light2 Television
Kitchen Device2 Light1 Light2 Microwave Fan1
Lobby Device3 Light1 Light7 Fan1 Fan8
Device3
updated
```

Verify with the database



```
Reading table information for completion of table and column names
You can turn off this feature to get a quicker startup with -A

Database changed
MariaDB [SQL_GATEWAY_DATABASE]> SELECT * FROM CONTROLBOX_CONFIG;
+---------+---------+------------+------------+------------+------------+
| ROOM    | DEVICE  | APPLIANCE1 | APPLIANCE2 | APPLIANCE3 | APPLIANCE4 |
+---------+---------+------------+------------+------------+------------+
| Hall    | Device1 | Fan2       | Light1     | Light2     | Television |
| Kitchen | Device2 | Light1     | Light2     | Microwave  | Fan1       |
+---------+---------+------------+------------+------------+------------+
2 rows in set (0.001 sec)

MariaDB [SQL_GATEWAY_DATABASE]> SELECT * FROM CONTROLBOX_CONFIG;
+---------+---------+------------+------------+------------+------------+
| ROOM    | DEVICE  | APPLIANCE1 | APPLIANCE2 | APPLIANCE3 | APPLIANCE4 |
+---------+---------+------------+------------+------------+------------+
| Hall    | Device1 | Fan2       | Light1     | Light2     | Television |
| Kitchen | Device2 | Light1     | Light2     | Microwave  | Fan1       |
| Lobby   | Device3 | Light1     | Light7     | Fan1       | Fan8       |
+---------+---------+------------+------------+------------+------------+
3 rows in set (0.000 sec)

MariaDB [SQL_GATEWAY_DATABASE]>
```

```
inserted
updated
{"Room":"Lobby","Device":"Device3","Appliance1":"Light1","Appliance2":"Light7","Appliance3":"Fan1","Appliance4":"Fan8"}
Lobby Device3 Light1 Light7 Fan1 Fan8
came
Hall Device1 Fan2 Light1 Light2 Television
Kitchen Device2 Light1 Light2 Microwave Fan1
Lobby Device3 Light1 Light7 Fan1 Fan8
Device3
updated
```

5. Mobile - Gateway – ControlBox

Control appliance

Step 1- To ON/OFF appliance User will send a JSON message with topic Room/ControlBox

JSON format will be-

{ “Username”:”abcd”, “AccessKey”:”xyz”, “Location”:”xyz”
“ApplianceName”:”Appliance”, “state”:”ON/OFF”}

Or

{ “Username”:”abcd”, “AccessKey”:”xyz”, “Location”:”xyz”, “ControlBox”:
“BoxName”, “ApplianceName”:”appliance”, “speed”:”0/1/2…”}

**** Here if speed is 0, means appliance is in OFF condition**

Step 2- The appliance will be ON or OFF based on message also device will send an acknowledgement message to user that their command has been executed successfully. Acknowledgement message will be published with topic Acknowledgement/Room/ControlBox.

JSON message will be- { “Username”:”abcd”, “AccessKey”:”xyz”,
“Location”:”xyz” “ApplianceName”:”Appliance”,
“state”:”ON/OFF”}

Or
{ “Username”:”abcd”, “AccessKey”:”xyz”, “Location”:”xyz”, “ControlBox”:
“BoxName”, “ApplianceName”:”appliance”, “speed”:”0/1/2…”}

Step 3-When device will send the acknowledgement message, gateway will save the current device state in table named Room_ControlBox_Appliance. And the columns in the table will be-

Date	Location	Client	Appliance	ON time	OFF time	Total ON time

Update security key (admin user only)

Step 1- **Only admin can update the security key. To update the security key admin need to open their app and go into gateway setting.**

Step 2- There will be an option to update security key, when user click on the option, gateway will response with a webpage secured with updated user id and password.

Step 3- User need to login into the webpage with its user id and password, if it is correct webpage will provide option to update security key.

Step 4- User will enter new security key and save it.

Step 5- Gateway will update the security key in MySQL table user_data.

Step 6- Since gateway is in sync with cloud, security will also be updated in cloud. Also, if you change security key in cloud it will updated in gateway.

VIDEO LINK-
https://drive.google.com/file/d/1arGjy2AjAy94Mkp11ctuQJ2PHv87-6wA/view?usp=sharing

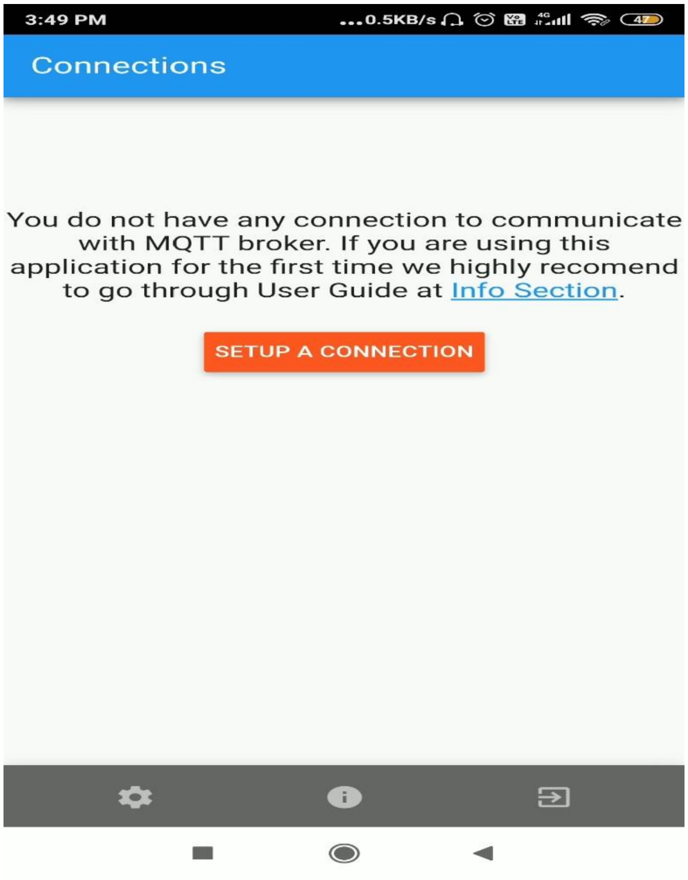
Json format

{"Username":"your_username", "AccessKey":"xyz", "Location":"Room_name", "Appliance":"Appliance_name", "state":"0/1"}

Example
{"Username":"username1", "AccessKey":"xyz", "Location":"Lobby", "Appliance":"Light1", "state":"5"}

Download the app
https://play.google.com/store/apps/details?id=snr.lab.iotmqttpanel.prod

open the app



3:49 PM ...0.5KB/s

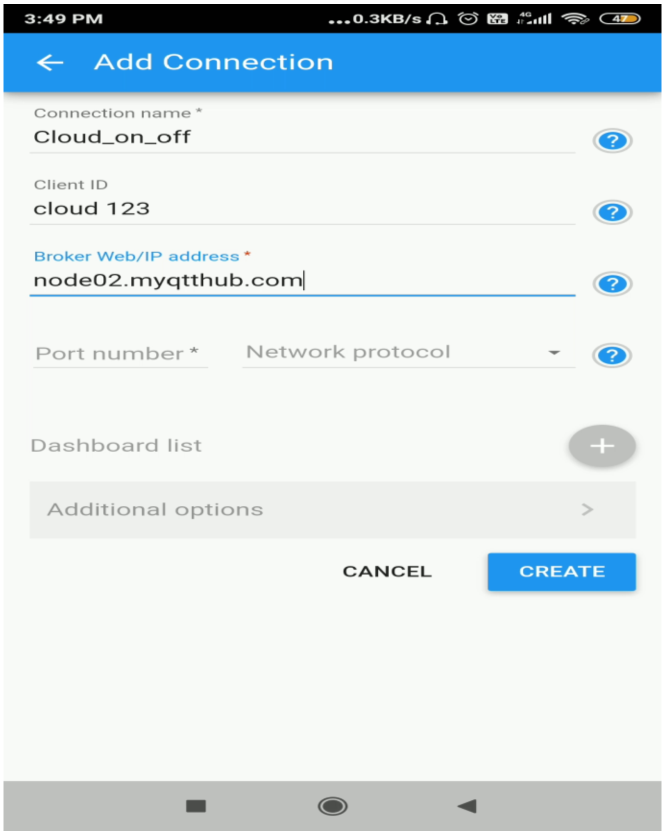
Connections

You do not have any connection to communicate with MQTT broker. If you are using this application for the first time we highly recomend to go through User Guide at Info Section.

SETUP A CONNECTION

Lets configure cloud details

first



3:49 PM ...0.3KB/s

Add Connection

Connection name *
Cloud_on_off

Client ID
cloud 123

Broker Web/IP address *
node02.myqtthub.com

Port number *

Network protocol

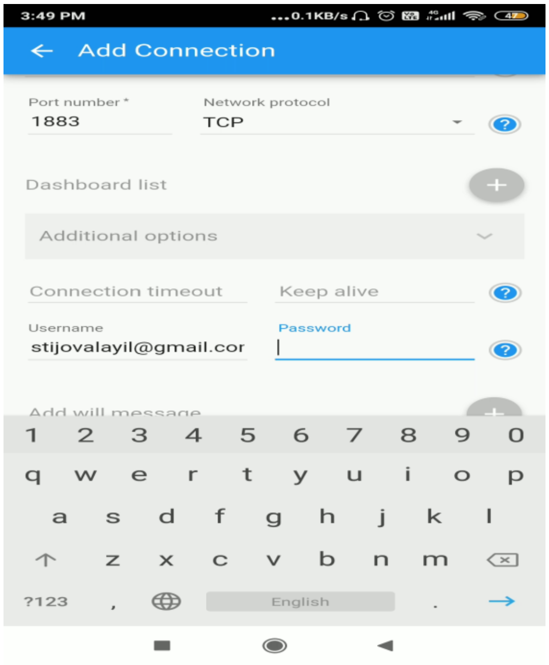
Dashboard list

Additional options

CANCEL

CREATE

Type your bootstrap credentials username and password here



Add Connection

Port number *: 1883

Network protocol: TCP

Dashboard list

Additional options

Connection timeout

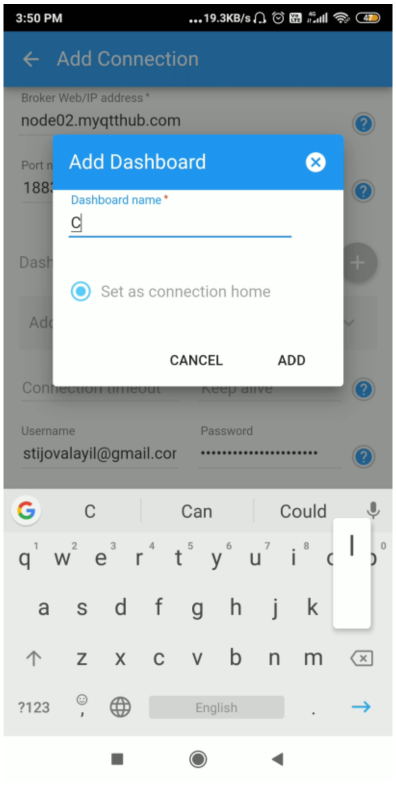
Keep alive

Username: stijovalayil@gmail.cor

Password:

Add will message

Add dashboard any name



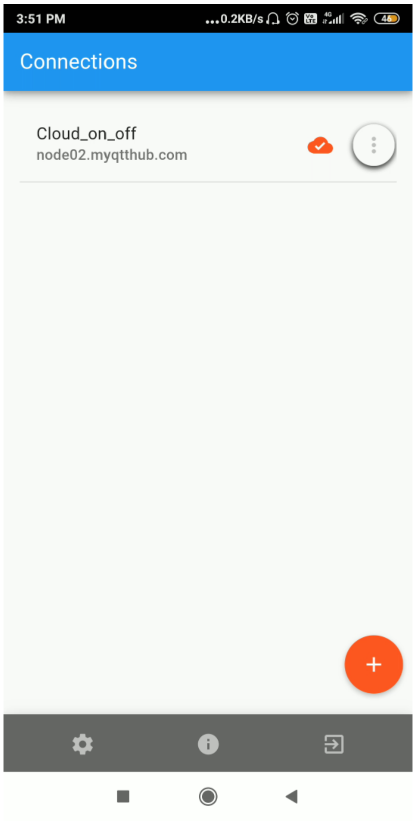
Broker Web/IP address *
node02.myqtthub.com

Dashboard name *
C

○ Set as connection home

Username
stijovalayil@gmail.cor

Password
••••••••••••••••••••••

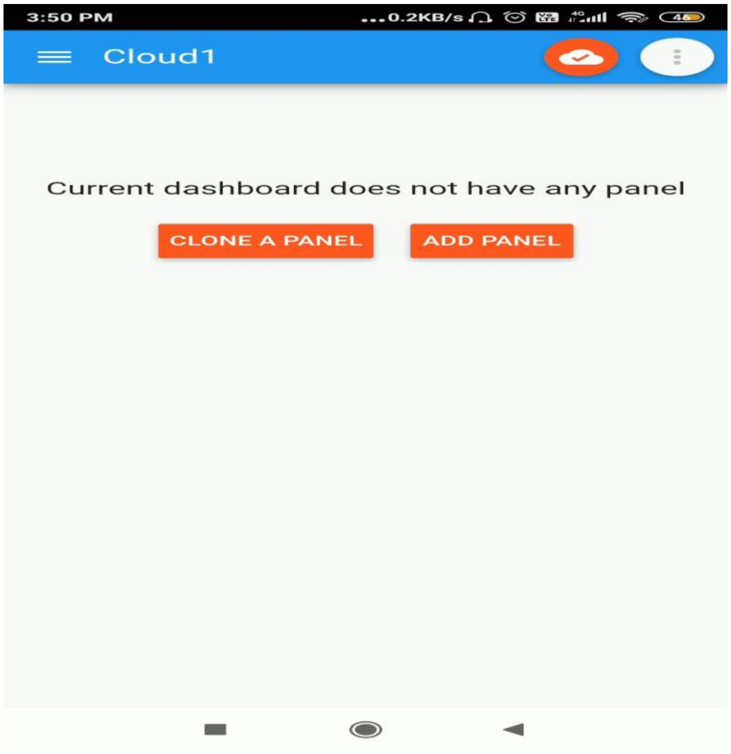


3:51 PM

Connections

Cloud_on_off
node02.myqtthub.com

Add panel

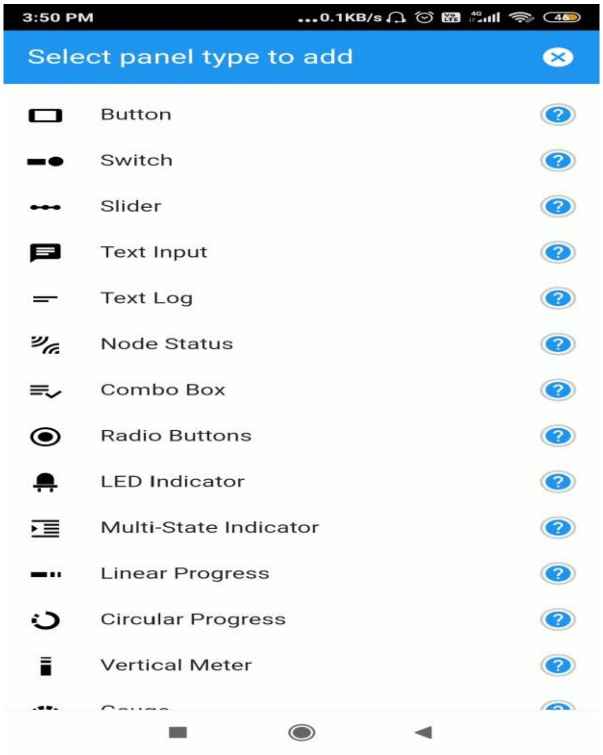


3:50 PM

Cloud1

Current dashboard does not have any panel

Press on switch



Topic

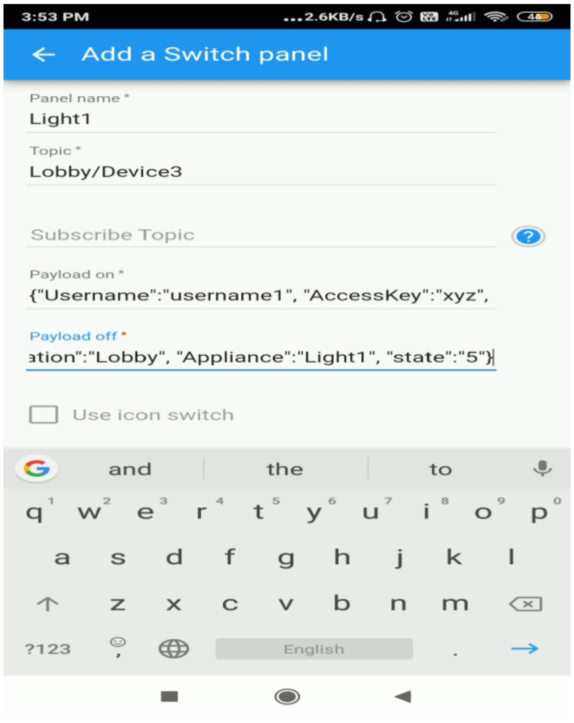
Lobby/Device3

Payload on

{"Username":"username1", "AccessKey":"xyz", "Location":"Lobby", "Appliance":"Light1", "state":"5"}

Payload off

{"Username":"username1", "AccessKey":"xyz", "Location":"Lobby", "Appliance":"Light1", "state":"0"}



Add a Switch panel

Panel name *
Light1

Topic *
Lobby/Device3

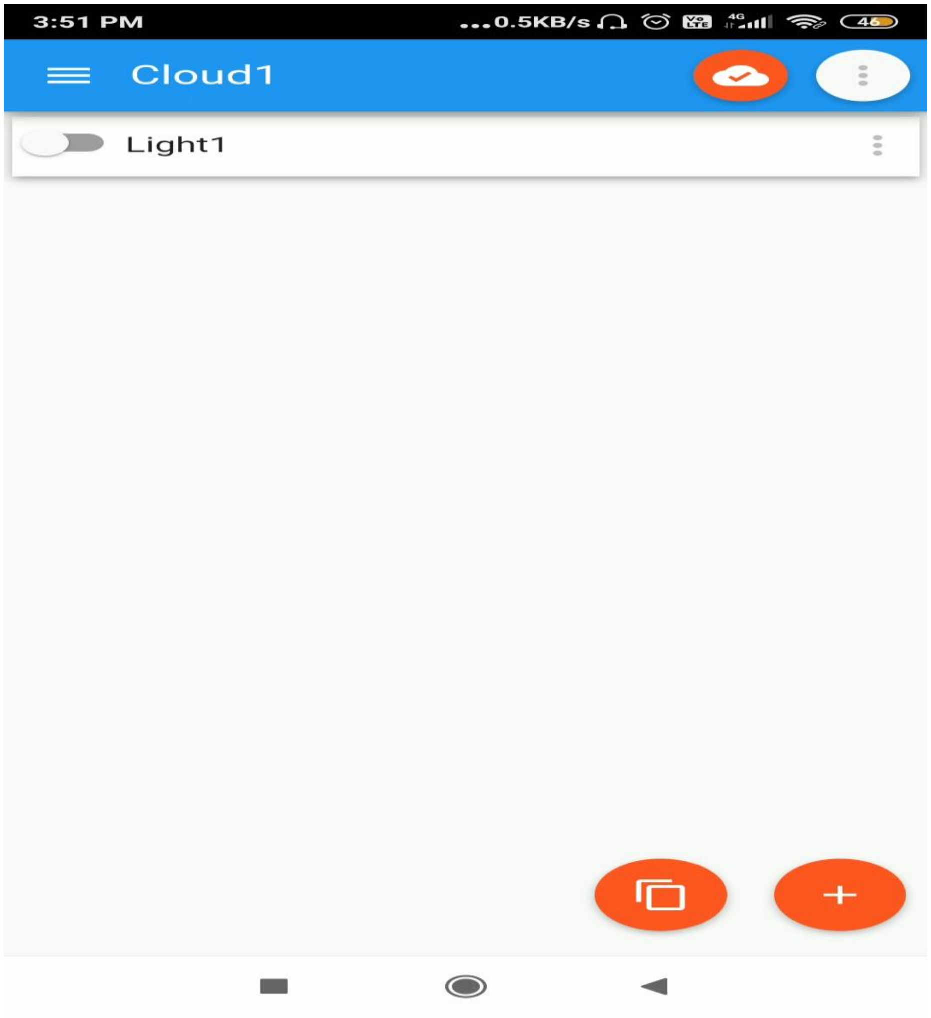
Subscribe Topic

Payload on *
{"Username":"username1", "AccessKey":"xyz",

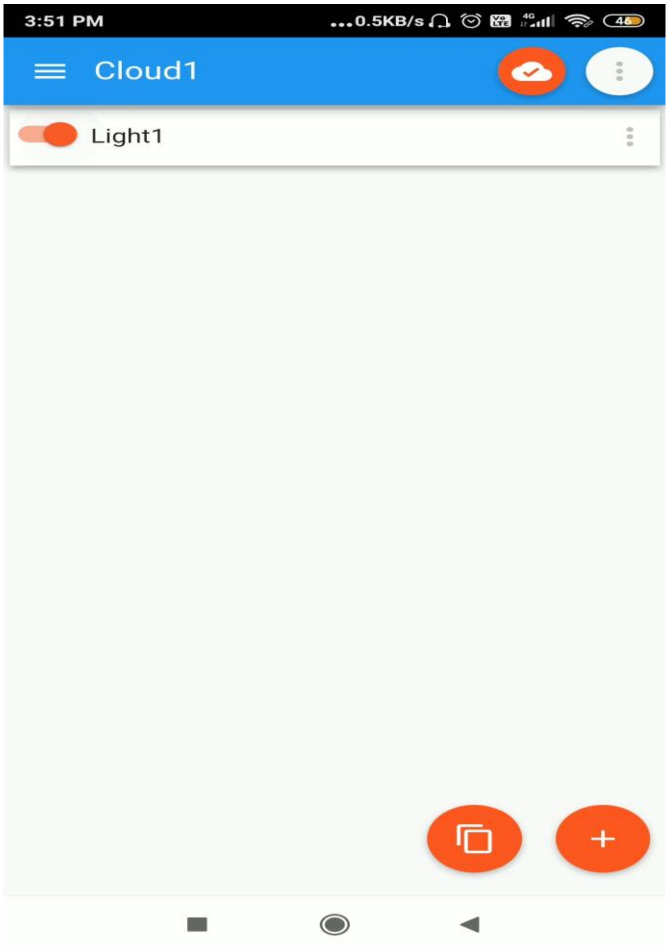
Payload off *
ation":"Lobby", "Appliance":"Light1", "state":"5"}

□ Use icon switch

save



Turn on light slide it



Light will turn on

Similarly configure for

local,

Port 1884

Username

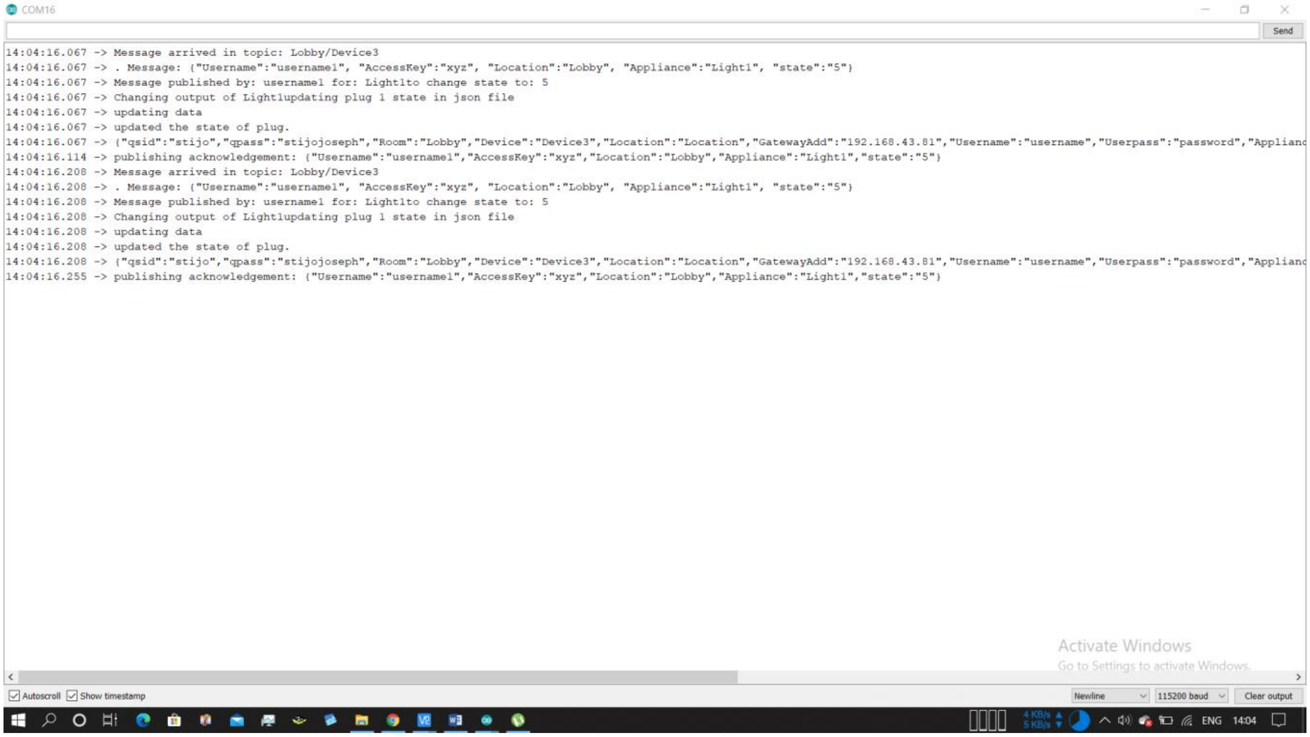
password   will be local

mosquito broker username

and password

check the serial monitor for

light1 turning on status



```
14:04:16.067 -> Message arrived in topic: Lobby/Device3
14:04:16.067 -> . Message: {"Username":"username1", "AccessKey":"xyz", "Location":"Lobby", "Appliance":"Light1", "state":"5"}
14:04:16.067 -> Message published by: username1 for: Light1to change state to: 5
14:04:16.067 -> Changing output of Light1updating plug 1 state in json file
14:04:16.067 -> updating data
14:04:16.067 -> updated the state of plug.
14:04:16.067 -> {"qsid":"stijo","qpass":"stijojoseph","Room":"Lobby","Device":"Device3","Location":"Location","GatewayAdd":"192.168.43.81","Username":"username","Userpass":"password","Applianc
14:04:16.114 -> publishing acknowledgement: {"Username":"username1","AccessKey":"xyz","Location":"Lobby","Appliance":"Light1","state":"5"}
14:04:16.208 -> Message arrived in topic: Lobby/Device3
14:04:16.208 -> . Message: {"Username":"username1", "AccessKey":"xyz", "Location":"Lobby", "Appliance":"Light1", "state":"5"}
14:04:16.208 -> Message published by: username1 for: Light1to change state to: 5
14:04:16.208 -> Changing output of Light1updating plug 1 state in json file
14:04:16.208 -> updating data
14:04:16.208 -> updated the state of plug.
14:04:16.208 -> {"qsid":"stijo","qpass":"stijojoseph","Room":"Lobby","Device":"Device3","Location":"Location","GatewayAdd":"192.168.43.81","Username":"username","Userpass":"password","Applianc
14:04:16.255 -> publishing acknowledgement: {"Username":"username1","AccessKey":"xyz","Location":"Lobby","Appliance":"Light1","state":"5"}
```

☑ Autoscroll ☑ Show timestamp

Newline

115200 baud

3. Mobile – Cloud

Account creation

Step 1- T

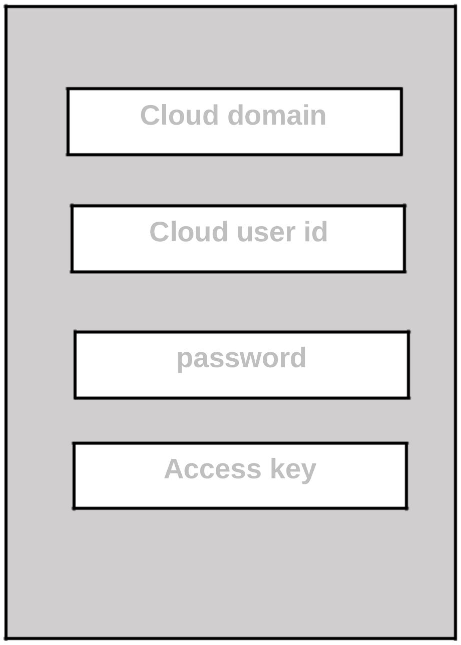
o access the gateway and control the devices user need to create an account in the cloud using app through a webpage.

Step 2- To create an account user need to provide some basic information like Name, Mobile no., e-mail, Location etc. and it will be verified by OTP.

Step 3- Now user will get a domain name, user id and password, access key from the cloud server.

Step 4- Again user come to the local mqtt broker (gateway) and choose option to connect with cloud in app.

Step 5- Gateway will ask for domain name, user id and password which was provided by cloud.



Save

Step 6- User will enter the above credential and click on save option.

Step 7- Using HTTP, Gateway will send a request to domain name with user id and password to connect with the cloud.

Step 8- If credentials provided by gateway is correct, it will connect with the cloud.

Step 9- Now cloud will ask gateway for saved data like, user_data, control box configuration, usage history etc. through a web page. Gateway will sync the data with the cloud.

3.Get appliance usage history and analytics For local and cloud

Step 1- To get the history of a appliance attached with a control box, admin User app will publish a message to gateway/cloud with Topic History/Room/ControlBox .

JSON message will be-

{ “Username”:”abcd”, “AccessKey”:”xyz”, “Location”:”xyz”,

“ControlBox”:”Boxname”, “state”:”History”, “StartDate”:”date”, “EndDate”:”date” }

Example:

{"Username":"username1", "AccessKey":"xyz","Location":

"Lobby","Appliance":"t1","from_date":"30/04/2021","to_date":"29/05/2021"}

Step 2- Since all the information present in the gateway is always in sync with cloud, cloud will response the user request.

Step 3- Cloud/gateway will retrieve the information from the MySQL database table Room_ControlBox_History.

Step 4- Cloud will send appliance usage history to the user using a webpage and app will show it accordingly

MODIFICATION: NOT WEBPAGE , HISTORY OF APPLIANCE IS PUBLISHED BACK IN JSON FORMAT THROUGH MQTT ITSELF

Msg received back, format

{"Username":"user_name" ,"AccessKey":"access_key","history":[{"DATE": "dateformat", "LOCATION": "room_name", "APPLIANCE": "appliance_name", "USERNAME": "username1", "ON_TIME": " time_format ", "OFF_TIME": "time_format", "TOTAL_ON_TIME": " time_format "}, {"DATE": " dateformat ", "LOCATION": " room_name ", "APPLIANCE": " appliance_name ", "USERNAME": "username1", "ON_TIME": " time_format ", "OFF_TIME": " time_format ", "TOTAL_ON_TIME": " time_format "}]}

{"Username":"username1" ,"AccessKey":"xyz","history":[{"DATE": "05/05/2021", "LOCATION": "Lobby", "APPLIANCE": "Light7", "USERNAME": "username1", "ON_TIME": "08:04:37", "OFF_TIME": "NULL", "TOTAL_ON_TIME": "NULL"}, {"DATE": "05/05/2021", "LOCATION": "Lobby", "APPLIANCE": "Light7", "USERNAME": "username1", "ON_TIME": "07:44:17", "OFF_TIME": "07:46:04", "TOTAL_ON_TIME": "0HRS2MINS"}]}

In here time_format is NULL in some places which means the appliance is not switched off yet
If appliance is not turned on or off anytime after the controlbox configurations and
The history request is given to database
The respone will be

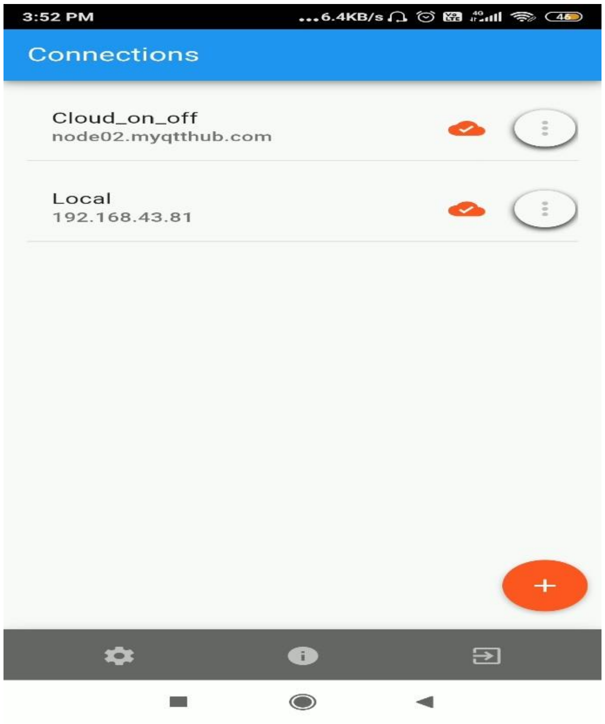
{"Username":"username1" ,"AccessKey":"xyz","history":[{"LOCATION": "Lobby", "APPLIANCE": "Fan2", "STATUS": "NO_HISTORY"}]}

Procedure setup on app

Select any cloud or local

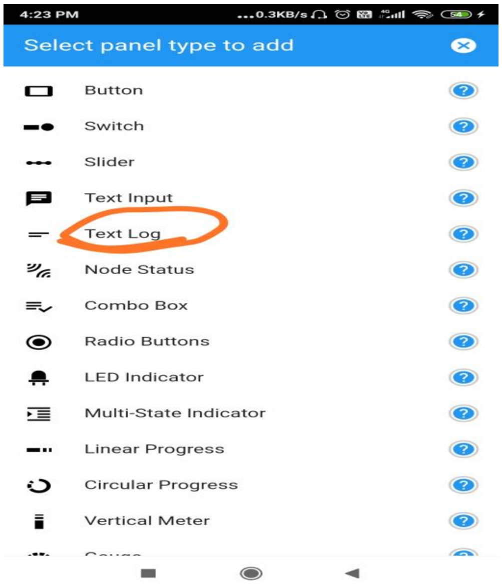
Video link -

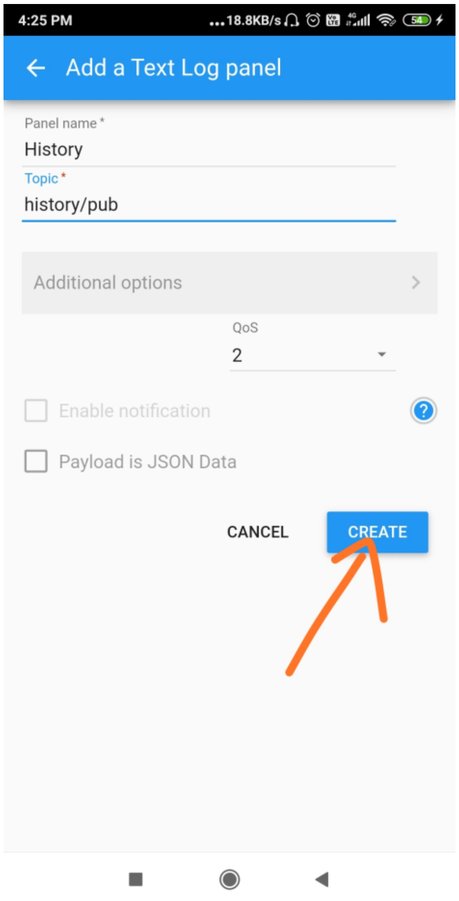
https://drive.google.com/file/d/1ppRlsmISWe8mO075NI9Pft9IptD5OKlF/view?usp=sharing



Cloud_on_off
node02.myqtthub.com

Local
192.168.43.81





Panel name *
History

Topic *
history/pub

QoS
2

□ Enable notification

□ Payload is JSON Data

Next for send the request to history of particular appliance setup

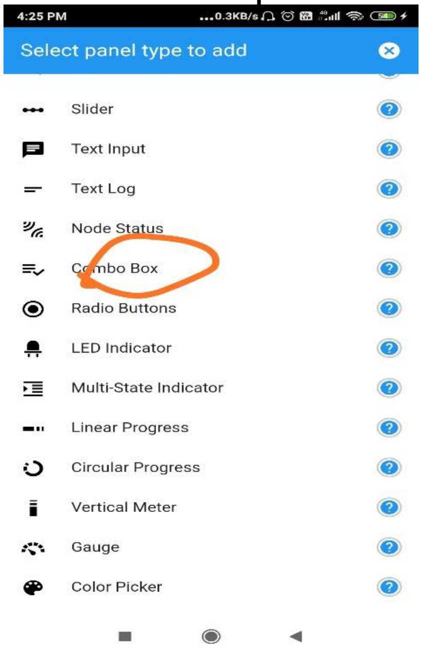
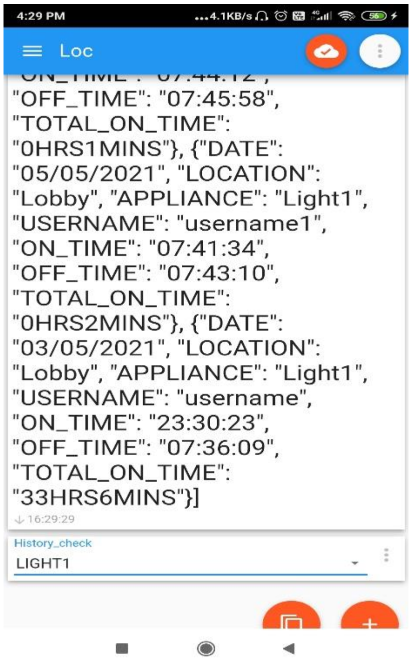


Image add in fututre

Final result



```
"ON_TIME": "07:44:12",
"OFF_TIME": "07:45:58",
"TOTAL_ON_TIME":
"0HRS1MINS"}, {"DATE":
"05/05/2021", "LOCATION":
"Lobby", "APPLIANCE": "Light1",
"USERNAME": "username1",
"ON_TIME": "07:41:34",
"OFF_TIME": "07:43:10",
"TOTAL_ON_TIME":
"0HRS2MINS"}, {"DATE":
"03/05/2021", "LOCATION":
"Lobby", "APPLIANCE": "Light1",
"USERNAME": "username",
"ON_TIME": "23:30:23",
"OFF_TIME": "07:36:09",
"TOTAL_ON_TIME":
"33HRS6MINS"}]
```

16:29:29

History_check: LIGHT1

4.Mobile - Cloud – Gateway Get Live Appliances Status

Get Live Appliances Status

Step 1- To get the live appliance status, user will publish a JSON format message to the control box with topic Room/ControlBox

JSON message will be –

{ “Username”:”abcd”, “AccessKey”:”xyz”, “Location”:”xyz”, “ApplianceName”:”Appliance”, “state”:”status”}

Example

{"Username":"username1", "AccessKey":"xyz", "location" : "Lobby","appliance" : "Light1"}

Step 2- Now control box will reply with appliance status with topic Status/Room/ControlBox.

JSON message will be-

{ “Username”:”abcd”, “AccessKey”:”xyz”, “Location”:”xyz” “ApplianceName”:”Appliance”, “state”:”ON/OFF”}

Or

{ “Username”:”abcd”, “AccessKey”:”xyz”, “Location”:”xyz”, “ControlBox”: “BoxName”, “ApplianceName”:”appliance”, “speed”:”0/1/2…”}

Step 3- App will show this received data.

FOR INDIVIDUAL REQUEST THE RESPONSE WILL BE

{"Username":"username1" ,"AccessKey":"xyz","status":{ "location":"Lobby","appliance":"Light1","status": "OFF"}}

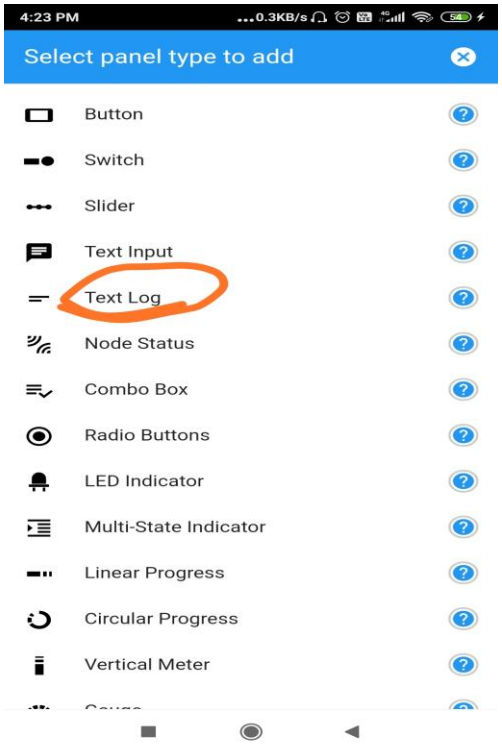
Or

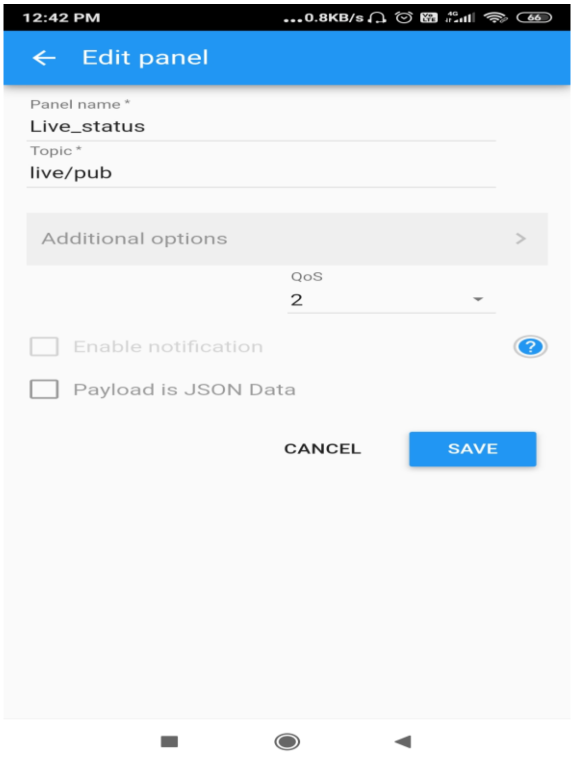
If appliance is equal to ALL

```
{"Username":"username1" ,"AccessKey":"xyz","status":[{"Device1": [{
"location":"Hall","appliance":"Fan2","status": "OFF"},{
"location":"Hall","appliance":"Light1","status": "OFF"},{
"location":"Hall","appliance":"Light2","status": "OFF"},{
"location":"Hall","appliance":"Television","status": "OFF"}]},{"Device2": [{
"location":"Kitchen","appliance":"Light1","status": "OFF"},{
"location":"Kitchen","appliance":"Light2","status": "OFF"},{
"location":"Kitchen","appliance":"Microwave","status": "OFF"},{
"location":"Kitchen","appliance":"Fan1","status": "OFF"}]},{"Device3": [{
"location":"Lobby","appliance":"Light1","status": "OFF"},{
"location":"Lobby","appliance":"Light7","status": "ON"},{
"location":"Lobby","appliance":"Fan1","status": "OFF"},{
"location":"Lobby","appliance":"Fan8","status": "OFF"}]}]}
```

Video link-
https://drive.google.com/file/d/1NtJzZTcrvPJpSnzoXvJLAwcv89MU4Yp2/view?usp=sharing

PROCEDURE TO SET UP PANEL





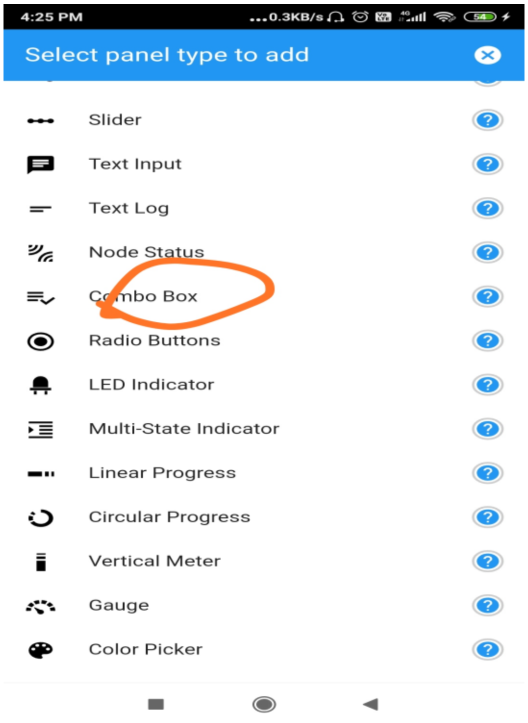
Panel name *
Live_status

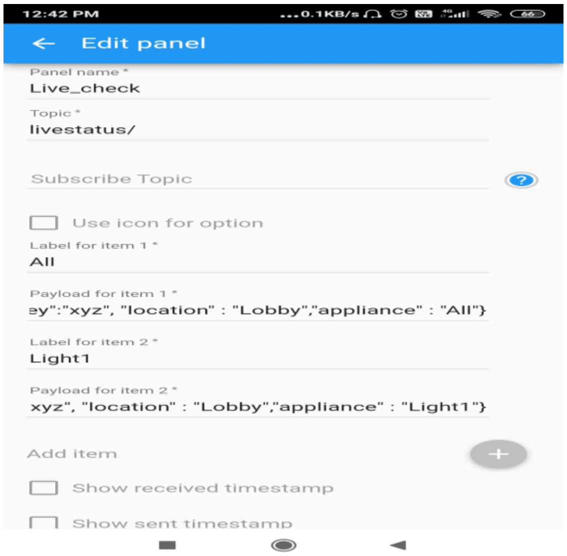
Topic *
live/pub

QoS
2

□ Enable notification

□ Payload is JSON Data





Panel name *
Live_check

Topic *
livestatus/

Subscribe Topic

□ Use icon for option

Label for item 1 *
All

Payload for item 1 *
ey":"xyz", "location" : "Lobby","appliance" : "All"}

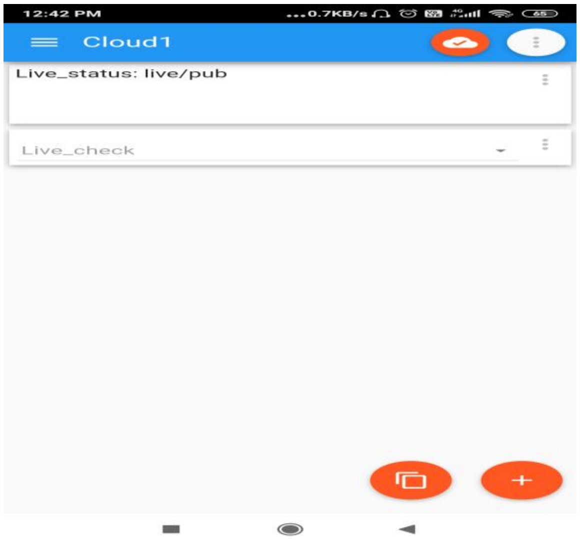
Label for item 2 *
Light1

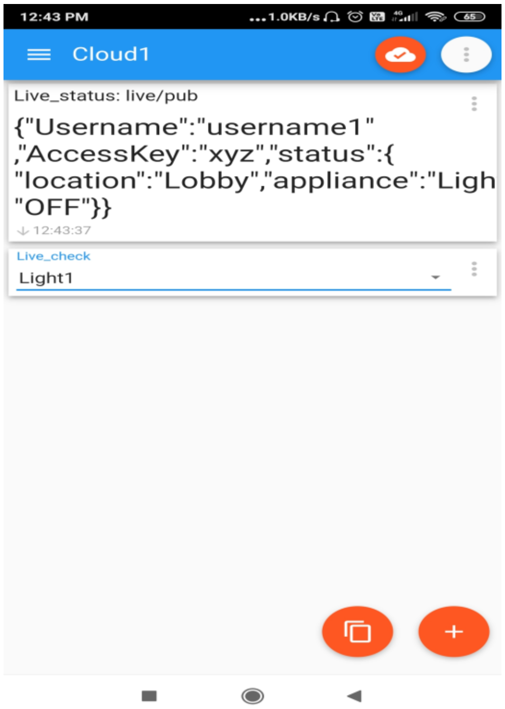
Payload for item 2 *
xyz", "location" : "Lobby","appliance" : "Light1"}

Add item +

□ Show received timestamp

□ Show sent timestamp





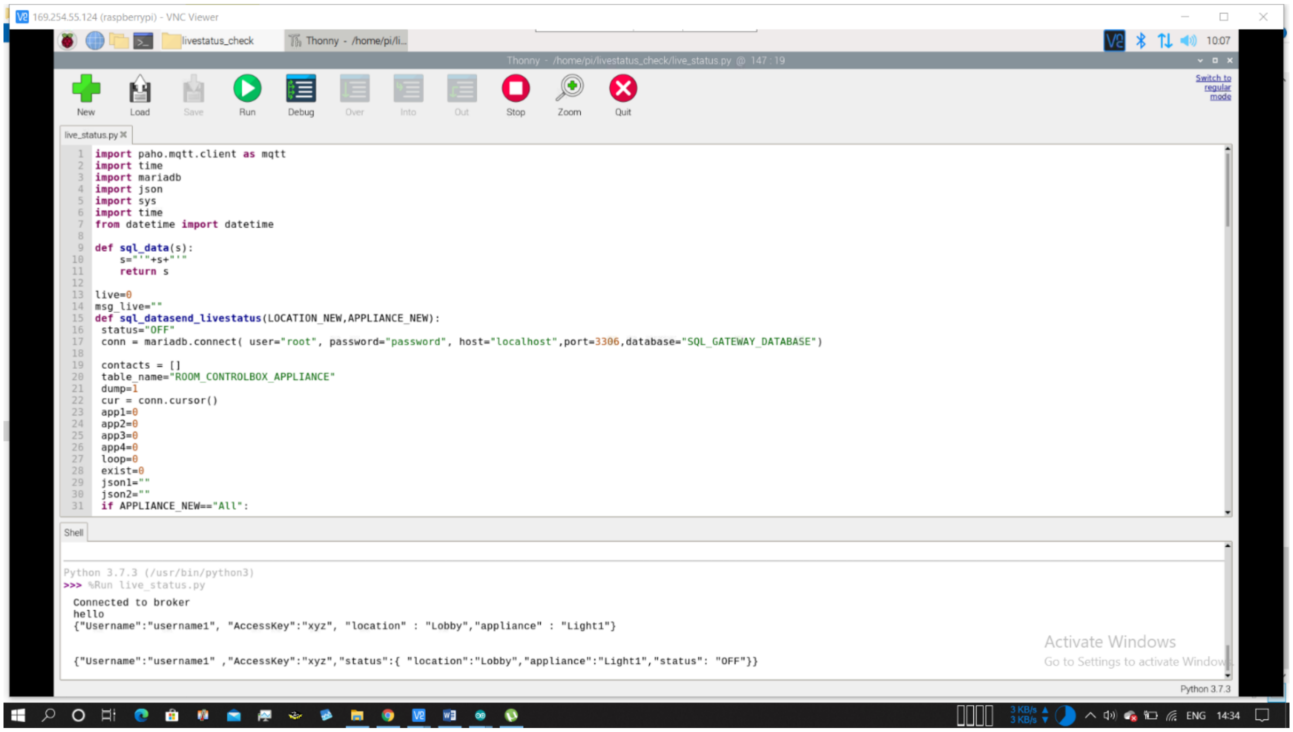
Live_status: live/pub

```
{"Username":"username1"
,"AccessKey":"xyz","status":{
"location":"Lobby","appliance":"Ligh
"OFF"}}
```

12:43:37

Live_check: Light1

OPEN LIVESTATUS_CHECCK FOLDER
EXECUTE THE livestatus.py file



```python
import paho.mqtt.client as mqtt
import time
import mariadb
import json
import sys
import time
from datetime import datetime

def sql_data(s):
    s="'"+s+"'"
    return s

live=0
msg_live=""
def sql_datasend_livestatus(LOCATION_NEW,APPLIANCE_NEW):
 status="OFF"
 conn = mariadb.connect( user="root", password="password", host="localhost",port=3306,database="SQL_GATEWAY_DATABASE")

 contacts = []
 table_name="ROOM_CONTROLBOX_APPLIANCE"
 dump=1
 cur = conn.cursor()
 app1=0
 app2=0
 app3=0
 app4=0
 loop=0
 exist=0
 json1=""
 json2=""
 if APPLIANCE_NEW=="All":
```

```
Python 3.7.3 (/usr/bin/python3)
>>> %Run live_status.py

 Connected to broker
 hello
 {"Username":"username1", "AccessKey":"xyz", "location" : "Lobby","appliance" : "Light1"}


 {"Username":"username1" ,"AccessKey":"xyz","status":{ "location":"Lobby","appliance":"Light1","status": "OFF"}}
```

NOTE:

AFTER VERIFYING ALL THESE PYTHON FILES AND CONFIGURATIONS

YOU CAN EXECUTE A SINGLE PYTHON FILE mainprogram.py
Which handles all the tasks which you have individually verified now

mainprogram.py

and sent the request from the panels in the app
you will get the result for everything
appliance on/off
live status
history

via cloud as well as local

and this python file we will be executing on boot

5. Mobile - Cloud - Gateway – Control

Live stream of camera

Step 1- User will turn ON the camera device, it will open in hotspot mode, user will provide Wi-Fi credential, camera name, gateway ip address.

Step 2- When app connects with gateway again, app will send camera configuration details to gateway, gateway will save its details in its MySQL table control_box_configuration.

Step 3- Camera device will now open in Wi-Fi mode and connect with wi-fi. Also connect with gateway using ip address.

Step 4- Now camera will live stream the video to the gateway.

Step 5- Gateway will store the video in a flash drive, also if user want see live stream, user will choose an option in app for live stream of camera.

Step 6- Gateway will response with a webpage of live streaming of camera.

Step 7- if user request for live stream on cloud, gateway will stream the video to cloud and cloud will stream the video to the user.

Step 8- The main advantage of storing and streaming video with gateway is, we can access older recordings, intruder detection, motion detection and face recognition can be done in gateway by using a python program.

7. ControlBox – Gateway

Activation of control box to gateway

Step 1- User will give the credential to control box as written in point 2. After this control box will be switched to Wi-Fi mode also app will connect with the wi-fi.

Step 2- Now app will request to gateway for the webpage to save the new control box configurations.

Step 3- If admin user want to create an appliance policy, admin user app will publish a JSON message with topic Gateway/AppliancePolicy

JSON message will be

{“UserName”:”xyz”, “AccessKey”:”xyz”, “Location”:”xyz”, “Room”:”xyz”, “ControlBox”:”xyz”, “ApplianceName”:”Appliance”, “TriggerTime”:”00:00:00”, “EndTime”:”00:00:00”, “PolicyName”:”abcd”}

Gateway will save the policy in the table name Policy_table

Policy Name	Client	Location	Date	Time	Room	Control Box	Trigger Time	End Time

Step 4- Now gateway will publish a message to the appliance on Trigger time with topic Room/ControlBox (Username will be gateway)

JSON message will be-

{ “Username”:”gateway”, “AccessKey”:”xyz”, “Location”:”xyz” “ApplianceName”:”Appliance”, “state”:”ON”}

Step 5- Control box will publish an acknowledgement message.

Step 6- On End time gateway will again publish a message with topic Room/ControlBox

JSON message will be-

{ “Username”:”gateway”, “AccessKey”:”xyz”, “Location”:”xyz” “ApplianceName”:”Appliance”, “state”:”OFF”}

Step 7- Control box will again send an acknowledgement message.

ALREADY IMPLEMENTED CLOUD, LIVESTATUS AND HISTORY GETTING ALONG WITH LOCAL CONFIGURATIONS

8. WebCloud – Gateway

Get gateway configurations

TO GET THE CONFIGURATION DETAILS AND HISTPR

Step 1- When user create an account with cloud, cloud will provide him a domain name, user id, password and access key.

Step 2- Now user will open app and click on gateway setting, app will request for webpage of gateway setting.

Step 3- Gateway will response with a webpage and app will show the webpage.

Step 4- Webpage will be secured with a user id, password which was created by user when he was configuring the gateway. User will enter its user id and password.

Step 5- In web page there will be an option connect with cloud.

Step 6- User will click on this option. It will ask for domain name, user id, password and access key.

Step 7- It will try to connect with cloud, If all the credential will be right then gateway will connect with the cloud.

Step 8- Now cloud will ask for the gateway configuration and other database files.

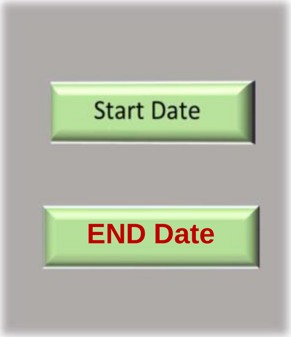
Step 9- Gateway will provide all the configuration and other information saved in MySQL database to the cloud. Or simply gateway will share the MySQL information with the cloud with a http web page.

Get appliance usage history

Step 1- **In app user will select the appliance for which he wants to get the usage history.**

Step 2- **App will request for a webpage which has access to table Room_ControlBox_Appliance in MySQL database.**

Step 3- **Cloud will response with a webpage having option Start date and End date.**



Step 4- User will enter the Start date DD/MM/YYYY and End Date DD/MM/YYYY.

Step 5- If cloud has usage history of the appliance it will directly send the information of usage history to the app otherwise it will again generate a request for the usage history with the gateway using http.

Step 6- Based on date python will retrieve information from table Room_ControlBox_Appliance in MySQL database and send it to cloud using webpage.

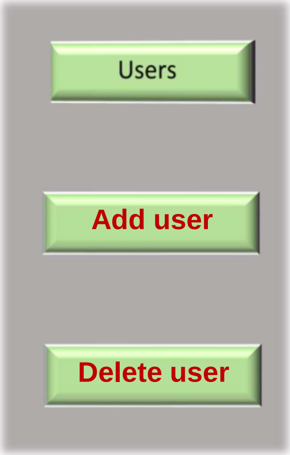
Step 7- Now cloud will again forward the information to the app using webpage.

Users data

Step 1- User will click on users option in the app. App will request for a webpage which has access to table user_data.

Step 2- Cloud will response with a webpage asking for user id and password.

Step 3- User will enter the information now again a web page will open with following options :



Step 4- User will click on first option i.e. Users.

Step 5- If cloud has user data it will directly send the information to the app using webpage otherwise it will again generate a request for the user data with the gateway using http.

Step 6- Now python will retrieve information from table user_data in MySQL database and send it to cloud using webpage.

Step 7- Now cloud will again forward the information to the app using webpage.

9. ControlBox - Gateway - Cloud - Mobile

Security Alarm notifications

Step 1- Any alarm like Water tank empty/full alarm, motion detection alarm, plant watering alarm etc. will be published by Topic Alarm/Room/ControlBox with a JSON message.

JSON message will be-

{“Appliance”:”WaterTank/MotionDetector….”, “AlarmType”:”Full/Empty/Detected/…..” }

Step 2- Gateway will receive the message and forward it cloud using mqtt bridge.

Step 3- Mobile should subscribe the Alarm topic. If mobile is connected with the gateway using wi-fi the message will reach directly and a notification pop up will show on the screen of mobile and also mobile will start ring.

Step 4- If mobile is connected with the cloud, the mqtt bridge will publish the message to the cloud, cloud broker will forward the message to the mobile.

Step 5- In case of motion detection alarm by camera, if an image or a short video clip (10-20 sec.) is to be sent via message, Camera will stream the image or short video clip to gateway using http. Gateway will save the image/video and forward it to the cloud using http.

Step 6- Cloud will send a request to the user app to download the image or video using webpage.

Step 7- App will download the image/video and inform the user about the alar